WORKING PAPER SERIES

# Consistent Time Window Assignments for Stochastic Multi-Depot Multi-Commodity Pickup and Delivery

Shohre Zehtabian/Marlin W. Ulmer

Working Paper No. 2/2023

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

FACULTY OF ECONOMICS
AND MANAGEMENT

Impressum (§ 5 TMG)
*Herausgeber:*
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

*Verantwortlich für diese Ausgabe:*
Shohre Zehtabian, Marlin W. Ulmer
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

*Bezug über den Herausgeber*
ISSN 1615-4274

# Consistent Time Window Assignments for Stochastic Multi-Depot Multi-Commodity Pickup and Delivery

Shohre Zehtabian
Department of Economics and Business Economics, Aarhus BSS, Aarhus University, Denmark, szehtabian@econ.au.dk

Marlin W Ulmer
Otto-von-Guericke Universität Magdeburg, 39016, Magdeburg, Germany, marlin.ulmer@ovgu.de

In this paper, we present the problem of assigning consistent time windows for the collection of multiple fresh products from local farmers and delivering them to distribution centers for consolidation and further distribution in a short agri-food supply chain with stochastic demand. We formulate the problem as a two-stage stochastic program. In the first stage, the time windows are assigned from a set of discrete time windows to farmers and in the second stage, after the demand is realized, the collection routes are planned by solving yet a newly introduced multi-depot multi-commodity team orienteering problem with soft time windows. The objective is to minimize the overall travel time and the time window violations. To solve our problem, we design a (heuristic) progressive hedging algorithm to decompose the deterministic equivalent problem into subproblems for a sampled set of demand scenarios and guide the scenarios toward consensus time windows. Through numerical experiments, we show the value of considering demand uncertainty over solving the deterministic expected value problem and the superiority of our approach over benchmarks when it comes to reducing the routing cost as well as the inconvenience for farmers.

*Key words*: Agri-food supply chains, Time window assignment, Consistency, Two-stage stochastic programming, Progressive hedging algorithm

## 1. Introduction

There is an increasing trend worldwide for schools, hospitals, and company canteens to source their groceries locally, if possible. Products are picked up at farmers and transported to local distribution centers or *food hubs* where the shipment to the local canteens takes place (Schmidt-Forth 2022, USDA 2022). The reasons for this new local development are diverse (Palacios-Argüello et al. 2020). First, buying locally can reduce the emissions and cost for transportation significantly. Second, the quality standards of the products can be controlled more easily, as can the working conditions for the farmers. Third, with a shorter supply chain, the food is usually fresher, and therefore, healthier. Fourth, it improves public perception and trust, as it supports individual local farmers instead of the big global players.

With this new trend new challenges occur. The supply chain becomes shorter, but at the same time more fragmented, as individual farmers may offer only a subset of groceries and in limited quantities. Thus, the regular collection and transportation process of products from farmers to the local distribution centers is not trivial and becomes an important cost factor, also given the relatively high salaries for truck drivers. Furthermore, the smaller local suppliers are often responsible for the entire process of farming and handling the shipping. Therefore, a seamless and reliable operation is vital, to ensure that the farmers can work effectively. Here, timing is of particular importance. Ideally, the time of the day when groceries are picked up does not vary because the farmers need to ensure that the products are ready for pickup and at the same time, they need to follow their already stressful daily routines without many additional interruptions (Truchot and Andela 2018). The importance of this time consistency is amplified when considering that farmers may provide products for several distribution centers, all operating their own vehicles, resulting in multiple pickups per day. The goal of a company operating the food hubs is therefore to determine a consistent time window (TW) for each farmer that will allow efficient transportation while keeping TW-violations at a minimum.

Setting TWs is challenging for a variety of reasons. Vehicles often visit several farmers per trip to collect different products. Furthermore, vehicles from different distribution centers may visit the same farmer on the same day. Thus, when setting TWs for farmers, the routing of the fleet for all distribution centers has to be considered. While this is already a challenging optimization problem, the difficulty is increased by demand uncertainty at distribution centers. There might be days with low demand while on other days the demand might be higher than expected. Therefore, the TW-decisions have to account for varying demand scenarios and consequently, for different daily routing solutions. The resulting decision process is a combination of decisions made at the tactical and the operational level of the planning.

We model the problem as a two-stage stochastic program where in the first stage, TW-decisions are made. In the second stage, once the demand is realized, the daily collection and routing decisions are made with the goal of minimizing the routing costs and TW-violations simultaneously. The second stage can therefore be seen as a multi-depot multi-commodity orienteering problem with soft TWs. As solving the full deterministic equivalent problem for several scenarios of realistic sizes is computationally intractable, we design a heuristic solution approach based on a scenario decomposition technique; namely the progressive hedging algorithm (PHA) of Rockafellar and Wets (1991). Over a number of iterations, our (heuristic) PHA solves the individual scenarios and derives a consensus first-stage solution that is fed in the next iteration of PHA. Over time, the weight of the current consensus solution is increased, likely leading to convergence to a common solution for all scenarios. As the second-stage decisions are by themselves very challenging, we rely on a matheuristic. Our approach solves a route-based formulation of the second-stage multi-depot multi-commodity orienteering problem with soft TWs via a commercial solver by heuristically generating a pool of routes based on demand scenarios.

We test our method for a variety of instance settings to analyze both methodology and problem. We derive the following main insights:

- For small instances, our PHA provides solutions very close to optimality. For instances of real-world size, it outperforms other scenario-based methods and heuristic policies for all instances.
- Compared to optimizing on expected demands, our policy reduces both travel time and supplier inconvenience at the same time.
- "Soft" TW-consistency with rare and minor violations can be achieved at a cost increase of about 3%. Guaranteed TW-consistency increases overall routing cost by about 7% in our setting.
- Wider TWs can keep the cost of consistency reasonable while narrower TWs can become relatively costly.

The paper is organized as follows. In Section 2, we present the relevant literature. The problem is defined in Section 3. Our heuristic PHA is introduced in Section 4. In Section 5, we present the design of the experiments followed by the results in Section 6. The paper concludes with a summary and outlook in Section 7.

## 2. Literature Review

In the following, we discuss the related literature. Our work considers the consistent provision of TWs for farmers in an agri-food supply chain. First, we discuss the most relevant works and then provide details on the two individual domains.

### 2.1. Most Relevant Studies

Our problem shows similarities with the TW-assignment vehicle routing problem (TWAVRP). This problem was first introduced by Spliet and Gabor (2015) for a distribution network in which deliveries of stochastic demands from a single depot occur during prearranged TWs on a regular basis. TWs are selected with fixed lengths, before the customers' demands become known, in the first stage. Then, the routes are planned associated with a finite set of demand scenarios, respecting the assigned TWs in the second stage, while minimizing the expected transportation costs. Spliet and Gabor (2015) formulate the problem as a two-stage stochastic mixed integer linear program and design an exact branch-price-cut algorithm to solve it. Following this work, Spliet and Desaulniers (2015) present a discrete version of TWAVRP in which, associated with each customer, there is a finite set of predetermined TWs to select and assign one TW to the customer in the first stage. Similar to the original paper, Spliet and Desaulniers (2015) solve a deterministic set-covering formulation of the discrete TWAVRP with an exact method. In another follow-up study, Dalmeijer and Spliet (2018) solve a new formulation of TWAVRP with a faster branch-and-cut algorithm than the exact method in Spliet and Desaulniers (2015). TWAVPR with time-dependent travel times is introduced by Spliet, Dabia, and Van Woensel (2018) who then present a branch-price-cut algorithm for solving the instances from Spliet and Gabor (2015) with the addition of time-dependent travel times. As an improvement over the exact methods by Spliet and Gabor (2015) and Dalmeijer and Spliet (2018), Dalmeijer and Desaulniers (2021) design a new branch-price-cut algorithm for solving TWAVRP. Subramanyam, Wang, and Gounaris (2018) develop a branch-and-bound tree based scenario decomposition algorithm to solve TWAVRP and discrete TWAVRP benchmark instances from Spliet and Gabor (2015), Spliet and Desaulniers (2015), Dalmeijer and Spliet (2018), plus their own additional benchmark instances with a larger number of scenarios. Martins et al. (2019) also consider a similar problem to TWAVRP in which TWs are assigned to each product segment in the context of a multi-compartment vehicle routing problem. However, instead of considering demand scenarios for finding consistent TWs, they assume a multi-period planning horizon. Another extension to TWAVRP is the paper by Jalilvand, Bashiri, and Nikzad (2021) in which the authors solve the smallest instances from Spliet and Gabor (2015) with added stochastic service times by using a PHA.

Unlike all these studies, our problem at hand is assigning TWs for the collection of multiple products from farmers in the context of agri-food supply chains where the stochastic demand is at the distribution centers, meaning at the end of the routes. Thus, while the first stage decision is similar, the second stage differs significantly. Instead of a vehicle routing problem with TWs, a multi-commodity multi-depot orienteering problem with soft TWs has to be solved. Especially having multiple depots is a characteristic known for making routing problems more challenging to deal with (see the survey paper Montoya-Torres et al. (2015)).

Another related work is presented by Neves-Moreira et al. (2018). In the context of food retail, Neves-Moreira et al. (2018) present a problem with stochastic demand and product-segment dependent TWs for a distribution network in which products are delivered from a warehouse to retail sites within the assigned TWs. Neves-Moreira et al. (2018) solve their problem with a matheuristic which consists of a three-phase process of generating routes, solving a set-partitioning formulation of the problem, and improving the routes. Unlike our approach, the routes generated in Neves-Moreira et al. (2018) are restricted to direct trips with only one product, multi-location routes that deliver the same product to all locations, or multi-product direct trips. Our underlying problem also deals with multiple products. However, the TWs are assigned independently of the products/commodities.

### 2.2. Vehicle Routing for Agri-food Supply Chains

It is known that planning of the transportation of fresh products and agri-food grains from farmers to end customers via local intermediary distribution centers or food hubs is an important aspect of short agri-food supply chains (AFSCs). Transporting agri-food products has also attracted more attention from both academia and industry as well as public authorities in recent years (Prajapati et al. 2022). Food hubs are a special type of food distribution infrastructures where the collected supplies (food products) from local/regional farmers are consolidated (or sometimes processed) and then distributed to businesses such as grocery retailers and supermarkets, food and catering services, or to institutional kitchens, e.g. school canteens (Palacios-Argüello et al. 2020). In spite of all the attention given and changes applied to the local AFSCs, due to the ever-increasing demand for (locally and sustainably grown) agricultural products (Yadav et al. 2022), there are still issues that need to be addressed, such as how to manage the collection and distribution of the food, trucks, and TWs (Prajapati et al. 2022). In their systematic literature review of AFSCs, Yadav et al. (2022) list inefficient transportation as one of the main contributors to food waste/loss. Moreover, they mention the presence of uncertainty in upstream and downstream stages of ASFCs as a significant challenge in practical operations and in decision making processes. By looking into the literature, one can see that the vehicle routing problem (VRP) has been one of the less explored aspects of (short) AFSCs in academia (Yadav et al. 2022, Gu et al. 2021).

One of the few studies with the focus on VRP for AFSCs, which also provides an overview of the recent literature on e-commerce AFSCs, is the work by Prajapati et al. (2022). The authors investigate the problem of designing first-mile pickup of multiple products (food grains) from farmers in the rural areas, transporting them to local distribution centers, and last-mile delivery of those products from the centers to urban businesses. Their underlying framework is a two-echelon distribution problem in which a central e-commerce platform makes the decisions regarding the collection and delivery routes and trucks with the goal of minimizing transportation, product damage, and carbon emission costs as well as penalizing the late pickups from distribution centers and delays on deliveries to end customers. Gu et al. (2021) solve a multi-commodity two-echelon distribution problem in the context of a short and local fresh product supply chain in which farmers transport their supply to intermediary distribution centers with their own trucks. The centers consolidate the products and deliver them to end users (school canteens and supermarkets) with a fleet of homogeneous vehicles stationed at the centers. A central decision maker manages the distribution centers and designs collection and delivery routes with the goal of minimizing the total transportation cost of the network. In a smaller scale and only one echelon, Palomo-Martínez, Salazar-Aguilar, and Laporte (2017) study the problem of daily distribution of fresh fruits from a single supplier to local customers via an in-house heterogeneous fleet of vehicles. In their underlying framework, it is possible to have more demand than supply (and consequently not satisfying all customers' demands fully), plus to split the demand for a product among the vehicles. The authors model their problem as a multi-product split delivery capacitated team orienteering problem with the objective of maximizing the profit from satisfying the demand. In the context of short AFSC, our problem also considers split pickup and delivery (surveyed in Archetti and Speranza (2012), Mor and Speranza (2022)) of multiple commodities (see the survey paper by Archetti, Campbell, and Speranza (2016)). These characteristics are known to make the routing problems harder to solve (Archetti, Campbell, and Speranza 2016). Besides the discussed differences in the routing models, none of the aforementioned work considers uncertain demand as well as time consistency.

One important aspect of designing the distribution system of an AFSC is the policies for collecting products from farmers. One policy is to let farmers (suppliers) bring their products to food hubs. The work by Gu et al. (2021) is an example that uses this approach for collecting the commodities from suppliers in their two-echelon distribution framework. Another collection policy is to have coordinated pickups of products from farmers and delivering them to food hubs via routes (Palacios-Argüello et al. 2020). In route analysis for a network of local food supply chains, Bosona and Gebresenbet (2011) compare the two policies and show that using the second policy reduces the number of routes and the driving distance and total time of the routes. In our underlying problem, we use the latter policy.

### 2.3. Consistency in Vehicle Routing

In our work, we search for consistent TWs to allow efficient routing in the presence of uncertain demand. Consistency plays an increasingly important role in a variety of routing problems, as consistency not only gains the trust of customers in the service but also allows for drivers to perform less error-prone and better service with regular schedules (see Kovacs et al. (2014) for an overview). Besides transportation and delivery routing, consistency plays an important role in healthcare services where patients benefit from a regular schedule of visits, ideally from the same nurse. In general, consistency can assume several forms and can concern customers and drivers. From a customer's perspective, consistency may be in the time a service takes place or in the driver performing the service (see, e.g., Haughton (2007), Spliet and Dekker (2016), Song et al. (2020)). In this paper, we focus on the former aspect. Respectively, for drivers, consistency might be in the customers to visit, the areas to perform service (Zhong, Hall, and Dessouky 2007, Haugland, Ho, and Laporte 2007, Carlsson and Delage 2013), or the daily routes they travel (Sungur et al. 2010, Ackva and Ulmer 2022).

Ensuring consistency can be challenging for a variety of reasons. One of the most common causes is uncertainty in the service process, as we discuss in the following. However, even if everything is deterministic, finding consistent solutions might be challenging, for example, not every customer needs to be visited every day (see, e.g., Subramanyam and Gounaris (2018), Campelo et al. (2019), Stavropoulou, Repoussis, and Tarantilis (2019)). Uncertainty usually manifests in the demand (Haughton 2007, Crainic et al. 2011, Spliet and Desaulniers 2015, Spliet and Gabor 2015, Spliet, Dabia, and Van Woensel 2018, Dalmeijer and Spliet 2018, Dalmeijer and Desaulniers 2021), in the requesting customers (Zhong, Hall, and Dessouky 2007, Haugland, Ho, and Laporte 2007, Carlsson and Delage 2013, Song et al. 2020, Ackva and Ulmer 2022), or in the travel times (Crainic et al. 2012, Jabali et al. 2015, Vareias, Repoussis, and Tarantilis 2019). In our work, we also consider uncertain demand, however different than in other problems. The demand uncertainty does not occur at the suppliers who need to have their commodities collected, but at the set of distribution centers. Therefore, not every supplier has to be visited in every scenario (and some might be visited more than once). The corresponding problem is therefore not a vehicle routing problem, but a team orienteering problem. Another difference is that we consider explicitly a set of multiple commodities which is not studied yet in the consistency literature.

From a modeling and methodology perspective, much of the consistency work under uncertainty considers two-stage stochastic models. In the first stage, the consistent part of the decision (TW-assignment or routing) is determined. In the second stage, the planning is done once the stochasticity realizes. Very few papers solve the problems exactly. Some works propose the use of heuristics while others rely on scenario approximation methods, as we propose in this paper. For example, Song et al. (2020) propose a multiple-scenario approach (MSA, Bent and Van Hentenryck 2004) to determine driver-customer assignments under uncertain customer requests. The idea of an MSA is to solve a set of scenarios individually and find the most similar solution amongst them via a consensus function. For a service network design problem with uncertain travel times, Crainic et al. (2011) propose a heuristic PHA (Rockafellar and Wets 1991) that iteratively solves the individual scenarios, but enforces convergence to a common solution via penalty terms. We adapt both MSA and PHA to the needs of our problem and compare their performance.

## 3. Problem Definition

In this section, we present our problem. We describe the problem, formulate it mathematically, and finally give an example.

### 3.1. Problem Description

We consider a problem where a company provides regional groceries (now called "commodities") to local businesses and municipalities (now called "customers") via a set of distribution centers. In regular time steps (e.g., days or weeks), the customers demand a set of commodities to be provided in the next time step. The demand volumes are uncertain until the time of the order. The company satisfies the demand by providing the requested commodities at the distribution center closest to the customers. Thus, customer demand can be aggregated as the demand at the distribution centers and individual customers can be neglected from here on.

The commodities are collected from a set of regional farmers (now called "suppliers"). The suppliers have a contract with the company that ensures availability of a specific amount of each commodity in each time step, even though the company may decide to collect less (or even nothing). The suppliers are distributed within the service region and collection can take place for every distribution center. Thus, multiple vehicles might visit a supplier in a time period. For collection, the company can draw on capacitated vehicles with fixed activation cost and variable routing cost, at every distribution center. As the collection of the commodities at a supplier requires significant work, it is necessary for the supplier to be present at the time of the collection. However, the suppliers have substantial daily working responsibilities, often far away from the delivery point. Thus, time consistency in the pickup of the commodities is of utmost importance for them, especially, since several vehicles might visit the suppliers from different distribution centers on the same day. That means that regardless the demand in a period and the collection routes by the vehicles, (soft) TW during which vehicles usually arrive during their daily shifts for collection stays the same for a supplier. Therefore, the company aims at assigning TWs to suppliers that can be satisfied well without any substantial violation while at the same time allowing for cost-efficient collection routes regardless of the daily demand realization.

In summary, the problem can be seen as a two-stage stochastic program. In the first stage, TWs are assigned to the suppliers. In the second stage, demand realizes and collection routes for each distribution center are determined. The overall objective is to minimize the expected routing cost plus the cost of violating the assigned TWs.

### 3.2. Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a complete graph. $\mathcal{V} = \mathcal{S} \cup \mathcal{D}$ is the set of vertices, where $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$ is the set of suppliers and $\mathcal{D} = \{|\mathcal{S}|+1, \ldots, |\mathcal{S}|+|\mathcal{D}|\}$ represents the set of distribution centers. $\mathcal{E} = \{(j, j')|j, j' \in \mathcal{S} \cup \mathcal{D}\}$ is the set of edges between suppliers and distribution centers. Let $t_{jj'}$ be the travel time on edge $(j, j')$. Time $t_{jj'}$ includes the service time at location $j$, if $j \in \mathcal{S}$. Transfers between distribution centers are prohibited, i.e. $t_{jj'} = \infty$ for $j, j' \in \mathcal{D}$. In the rest of the paper, we refer to vertices as locations and edges as links. The specifications associated with the two stakeholders involved in this problem are as follows.

Suppliers provide a set of commodities $\mathcal{M}$. The commodities are picked up and transported to the distribution centers. Each supplier $s \in \mathcal{S}$ offers a maximum available quantity $O_{sm} \geq 0$ of commodity $m \in \mathcal{M}$. Let $\mathcal{M}_s = \{m \in \mathcal{M} | O_{sm} > 0\}$ be the set of commodities that a supplier offers.

Associated with each supplier $s$, there is a set of potential TWs, denoted by $W_s$, for the collection of their commodities. Let $w = [\underline{w}, \overline{w}] \in W_s$ be a candidate TW with $\underline{w}$ the earliest time and $\overline{w}$ the latest time that a supplier can be visited. From $W_s$, a TW is assigned to each supplier in the first stage. The penalty for arriving early at a supplier, i.e. earliness penalty, is $c_e$ per time unit. Similarly, the penalty for arriving late at a supplier, i.e. lateness penalty, is $c_l$ per time unit.

Let $\bar{\Omega}$ be the set of all random events that may be realized in the second stage. Consider the random vector $\mathbf{D} = [\mathbf{D}_{dm}]_{d\in\mathcal{D},m\in\mathcal{M}}$ that defines the spread of demands for commodities at distribution centers with the assumption that each $\mathbf{D}_{dm}$ has a known probabilistic distribution. For a given realization $\omega \in \bar{\Omega}$, $D^\omega = [D^\omega_{dm}]_{d\in\mathcal{D},m\in\mathcal{M}}$ is the vector of demands with $D^\omega_{dm} \geq 0$ as the demand of distribution center $d$ for commodity $m$. A sufficient number of homogeneous trucks with capacity $Q$ are available for hire at each distribution center to perform the collection of commodities from the suppliers and transporting them to the distribution centers.

We assume that the commodities are packed in standardized containers, therefore the supply and demand are integer. The collection of commodities from suppliers to each distribution center is performed via routes that start from the distribution center, visit a set of suppliers (ideally, in their designated TWs) and end at the same distribution center. Therefore, a route in our underlying problem is a sequence of locations. The supply of a supplier can be transported by more than one route. Let $R$ be the set of routes. Associated with each route $r \in R$ there is a parameter $\gamma(r)$, which represents the distribution center from which the route starts and ends at. $\sigma_{sr}$ is a binary parameter indicating whether supplier $s \in \mathcal{S}$ is on route $r \in \mathcal{R}$. Let $\delta_{ijr}$ be an indicator showing if location $j \in \mathcal{S} \cup \mathcal{D}$ is a successor of location $i \in \mathcal{S} \cup \mathcal{D}$ on route $r \in \mathcal{R}$.

In order to formulate the problem as a two-stage stochastic programming problem with recourse (Birge and Louveaux 2011), let $x_{sw} \in \{0,1\}$ be the first-stage decision variables denoting whether a TW $w \in W_s$ is assigned to supplier $s$, for $s \in \mathcal{S}$. Our first-stage problem can be formulated as in (1)–(3).

$$\min \mathbb{E}_{\mathbf{D}}[\mathcal{Q}(x, D^\omega)] \tag{1}$$

$$\text{s.t.} \sum_{w\in W_s} x_{sw} = 1, \qquad s \in \mathcal{S} \tag{2}$$

$$x_{sw} \in \{0,1\}. \qquad w \in W_s, s \in \mathcal{S} \tag{3}$$

Objective function (1) minimizes the expected second-stage cost only, as there is no cost associated with the first stage. Constraints (2) guarantee that each supplier is assigned one TW. Constraints (3) make sure that the first-stage variables are binary. Given the first-stage decision

$x = [x_{sw}]_{s\in\mathcal{S},w\in W_s}$, the second-stage subproblem ((4)–(16)) chooses the collection routes and evaluates the cost $\mathcal{Q}(x, D^\omega)$ for the demand scenario $D^\omega$ associated with realization $\omega \in \bar{\Omega}$.

$$\mathcal{Q}(x, D^\omega) = \min \sum_{s\in\mathcal{S}} \sum_{r\in R} \sigma_{sr} \left[c_e e_{sr} + c_l l_{sr}\right] + \sum_{r\in R} \sum_{\substack{d\in\mathcal{D}:\\ \gamma(r)=d}} a_{dr} \tag{4}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{\substack{r\in R:\\ \gamma(r)=d}} \sum_{\substack{s\in\mathcal{S}:\\ m\in\mathcal{M}_s}} \sigma_{sr} q_{msr} \geq D^\omega_{dm}, && d\in\mathcal{D}, m\in\mathcal{M} && (5)\\
& \sum_{s\in\mathcal{S}} \sum_{m\in\mathcal{M}_s} \sigma_{sr} q_{msr} \leq Q\theta_r, && r\in R && (6)\\
& \sum_{r\in R} \sigma_{sr} q_{msr} \leq O_{sm}, && s\in\mathcal{S}, m\in\mathcal{M}_s && (7)\\
& \sigma_{sr} e_{sr} \geq \underline{w} x_{sw} - \sigma_{sr} a_{sr} - M(1-x_{sw}), && s\in\mathcal{S}, w=[\underline{w},\overline{w}]\in W_s, r\in\mathcal{R} && (8)\\
& \sigma_{sr} l_{sr} \geq \sigma_{sr} a_{sr} - \overline{w} x_{sw} - M(1-x_{sw}), && s\in\mathcal{S}, w=[\underline{w},\overline{w}]\in W_s, r\in\mathcal{R} && (9)\\
& \delta_{ss'r}[a_{sr} + t_{ss'} - a_{s'r}] \leq M(1-\theta_r), && s,s'\in\mathcal{S}, s\neq s', r\in\mathcal{R} && (10)\\
& \delta_{sdr}[a_{sr} + t_{sd} - a_{dr} + \tau] \leq M(1-\theta_r), && s\in\mathcal{S}, d\in\mathcal{D}, r\in\mathcal{R}: \gamma(r)=d && (11)\\
& \delta_{dsr}[t_{ds} - a_{sr}] \leq M(1-\theta_r), && s\in\mathcal{S}, d\in\mathcal{D}, r\in\mathcal{R}: \gamma(r)=d && (12)\\
& \theta_r \in \{0,1\}, && r\in R && (13)\\
& q_{msr} \in \mathbb{N}_{\geq 0}, && m\in\mathcal{M}_s, s\in\mathcal{S}, r\in R && (14)\\
& a_{jr} \in \mathbb{N}_{\geq 0}, && j\in\mathcal{S}\cup\mathcal{D}, r\in R && (15)\\
& e_{sr}, l_{sr} \in \mathbb{N}_{\geq 0}. && s\in\mathcal{S}, r\in R && (16)
\end{aligned}$$

Variables $\theta_r$, $q_{msr}$, $a_{jr}$, $e_{sr}$, and $l_{sr}$ are the second-stage (or recourse) decision variables. Variable $\theta_r$ decides on whether route $r$ should be selected or not. Variable $q_{msr}$ decides on the quantity of commodity $m$ picked up from supplier $s$ on route $r$. Variable $a_{sr}$ is the start-service time at supplier $s$ on route $r$ and $a_{dr}$ is the arrival time of route $r$ to distribution center $d$. Given the assigned TWs in the first stage, variables $e_{sr}$ and $l_{sr}$ decide on how much a route can be early or late at supplier $s$. Objective function (4) aims at minimizing the total penalty of the violation of the suppliers' assigned TWs on the selected routes as well as the arrival times of the routes at the distribution centers. Constraints (5) make sure that enough quantity of each commodity required by a distribution center is collected from suppliers and transported on routes to satisfy the realized demand of the distribution center. Constraints (6) impose that the total quantity of commodities transported on each route does not violate the truck capacity. Constraints (7) require that the total quantity of each commodity collected from every supplier on routes respects the maximum available quantity of the commodity at the supplier. Constraints (8)–(9) are soft TW-constraints.

**Table 1** **Notations-Parameters**

| Parameters | |
|---|---|
| $\mathcal{S}$ | The set of suppliers, index $s$ |
| $\mathcal{D}$ | The set of distribution centers, index $d$ |
| $\mathcal{V} = \mathcal{S} \cup \mathcal{D}$ | The set of locations, i.e. stakeholders |
| $\mathcal{E}$ | The set of links between locations in $\mathcal{V}$, index $(j, j')$ |
| $c_e$, $c_l$ | Earliness and lateness penalties |
| $t_{jj'}$ | The travel time on link $(j, j') \in \mathcal{E}$, including service time at location $j$, if $j \in \mathcal{S}$ |
| $\mathcal{M}$ | The set of commodities, index $m$ |
| $Q$ | The capacity of trucks transporting commodities from suppliers to distribution centers |
| $O_{sm} \geq 0$ | The maximum available quantity of commodity $m \in \mathcal{M}$ offered by supplier $s \in \mathcal{S}$ |
| $\mathcal{M}_s$ | The set of commodities with $O_{sm} > 0$ for supplier $s \in \mathcal{S}$ |
| $W_s$ | The set of potential TWs for supplier $s \in \mathcal{S}$, index $w$ |
| $w = [\underline{w}, \overline{w}] \in W_s$ | A candidate TW with $\underline{w}$ the earliest time and $\overline{w}$ the latest time to visit supplier $s \in \mathcal{S}$ |
| $\mathbf{D} = [\mathbf{D}_{dm}]_{d \in \mathcal{D}, m \in \mathcal{M}}$ | The random vector of demands of distribution centers for commodities |
| $\bar{\Omega}$ | The set of all random events that may be realized in the second stage, index $\omega$ |
| $D^\omega = [D^\omega_{dm}]_{d \in \mathcal{D}, m \in \mathcal{M}}$ | The vector of demands with $D^\omega_{dm} \geq 0$, for a given realization $\omega \in \bar{\Omega}$ |
| $R$ | The set of routes, index $r$ |
| $\gamma(r) \in \mathcal{D}$ | The distribution center from which route $r \in R$ starts and ends at |
| $\sigma_{sr}$ | A binary parameter indicating whether supplier $s \in \mathcal{S}$ is on route $r \in \mathcal{R}$ |
| $\delta_{jj'r}$ | A binary parameter indicating whether location $j'$ is a successor of location $j$, for $j, j' \in \mathcal{V}$, on route $r \in \mathcal{R}$ |
| $\tau \in \mathbb{R}_{\geq 0}$ | The activation cost of a truck, in time units |
| $M \in \mathbb{R}_{>0}$ | A large positive number associated with big-M constraints, e.g. greater than or equal to the travel time of the longest route plus $\tau$ |

These two sets of constraints link the second stage to the first stage. Constraints (10) make sure that the start-service time at a supplier succeeding another supplier on a route is planned no sooner than the time in which the previous supplier is served plus the travel time between the two. Constraints (11) enforce that the arrival time of a route at its distribution center to be after the last supplier on the route is served plus the travel time between the supplier and distribution center, and the activation cost of the truck in time units. Constraints (12) emphasize that the start-service time of the first supplier on a route is no sooner than the travel time between the distribution center and the supplier. Constraints (13)-(16) describe the second-stage variables. All notations and definitions are summarized in Tables 1 and 2.

### 3.3. An Illustrative Example

In the following, we give a small example to illustrate the two stages of the problem. An example of a graph is depicted in Figure 1. There are three suppliers, i.e. $\mathcal{S} = \{1, 2, 3\}$, who can provide two commodities. Supplier 1 can only provide maximum 15 units of commodity 1, supplier 2 has only a supply of maximum 10 units of commodity 2, and supplier 3 can offer maximum five units of

**Table 2 Notations-Variables**

| Variables | |
|---|---|
| $x_{sw} \in \{0,1\}$ | If TW $w \in W_s$ is assigned to supplier $s \in \mathcal{S}$: 1; otherwise: 0 |
| $\theta_r \in \{0,1\}$ | If route $r \in R$ is chosen: 1; otherwise: 0 |
| $q_{msr} \in \mathbb{N}_{\geq 0}$ | Quantity of commodity $m \in \mathcal{M}_s$ picked up at supplier $s \in \mathcal{S}$ on route $r \in R$ |
| $a_{sr} \in \mathbb{N}_{\geq 0}$ | Start-service time at supplier $s \in \mathcal{S}$ on route $r \in R$ |
| $a_{dr} \in \mathbb{N}_{\geq 0}$ | Arrival time at distribution center $d \in \mathcal{D}$ on route $r \in R$ (including truck activation cost) |
| $e_{sr} \in \mathbb{N}_{\geq 0}$ | Earliness at supplier $s \in \mathcal{S}$ on route $r \in R$ |
| $l_{sr} \in \mathbb{N}_{\geq 0}$ | Lateness at supplier $s \in \mathcal{S}$ on route $r \in R$ |

commodity 1 and 10 units of commodity 2. Suppliers in the figure are denoted by triangles with their supplies, i.e. $O_{sm}$, for each commodity are listed inside the triangles.

There are two distribution centers, i.e. $\mathcal{D} = \{4,5\}$, with stochastic demands $\mathbf{D}_{dm}$ for the two commodities. Two scenarios, $\omega_1$ and $\omega_2$, are considered with demand realizations for the distribution centers. In scenario $\omega_1$, the demand of distribution center 4 is composed of 15 units of commodity 1 and seven units of commodity 2. The demand of distribution 5 consists of 10 units of commodity 2. Therefore, the vector of demands associated with scenario $\omega_1$ is

$$D^{\omega_1} = \begin{bmatrix} D_{41}^{\omega_1} = 15 & D_{42}^{\omega_1} = 7 \\ D_{51}^{\omega_1} = 0 & D_{52}^{\omega_1} = 10 \end{bmatrix}.$$

On the other hand, in scenario $\omega_2$, the demand of distribution center 4 is only for 12 units of commodity 1. The demand of distribution center 5 is composed of five units of commodity 1 and 12 units of commodity 2. The vector of demands associated with scenario $\omega_2$ is

$$D^{\omega_2} = \begin{bmatrix} D_{41}^{\omega_2} = 12 & D_{42}^{\omega_2} = 0 \\ D_{51}^{\omega_2} = 5 & D_{52}^{\omega_2} = 12 \end{bmatrix}.$$

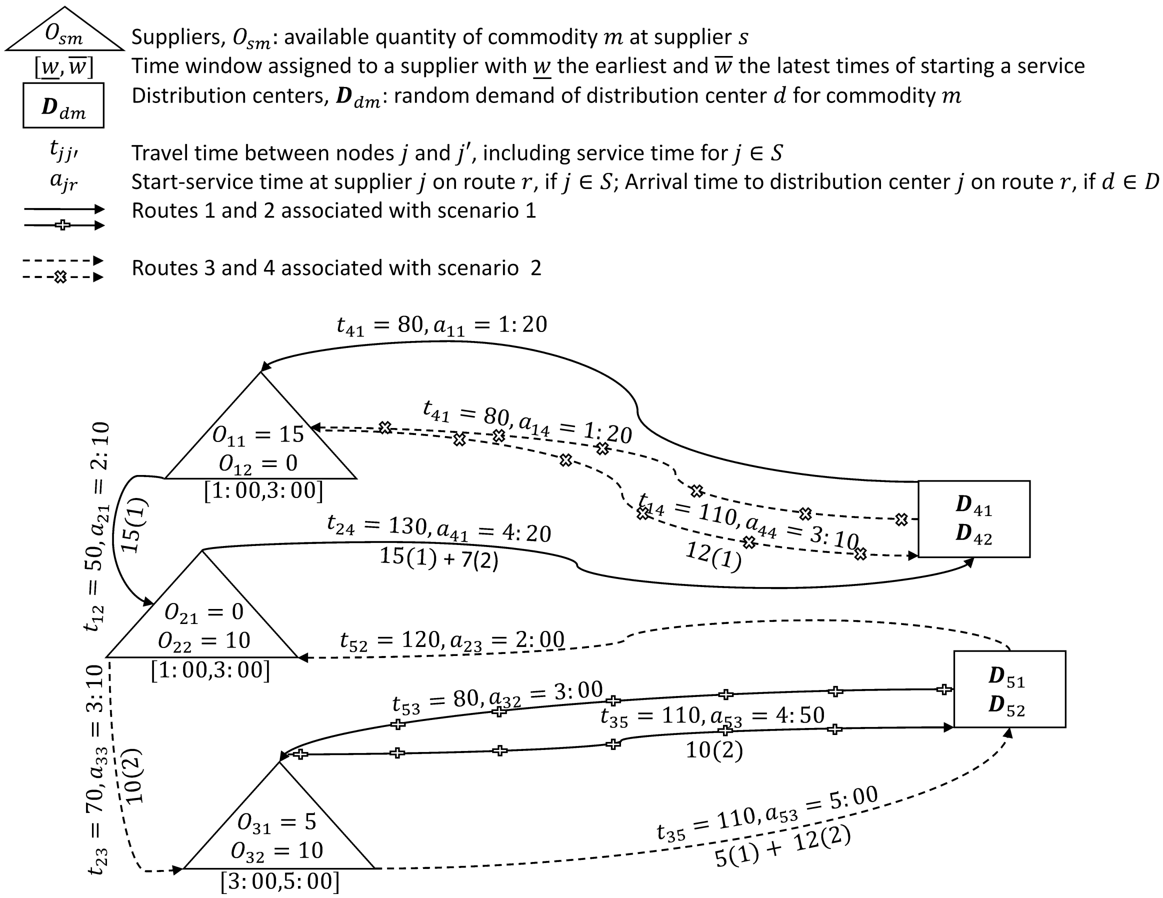
The distribution centers are depicted by rectangles in the figure. The travel time matrix of all links is as follows:

$$[t_{jj'}]_{|\mathcal{S}|\times|\mathcal{D}|} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 50 & 90 & 110 & 150 \\ 50 & 0 & 70 & 130 & 150 \\ 90 & 70 & 0 & 150 & 110 \\ 80 & 100 & 120 & 0 & \infty \\ 120 & 120 & 80 & \infty & 0 \end{pmatrix} \end{array}.$$

The travel time between two nodes includes 30 minutes service time, if the origin is a supplier. The activation cost of a truck, i.e. $\tau$, is assumed zero and the earliness and lateness penalties $c_e$ and $c_l$, are set to 100.

In the example, there is a set of two potential two-hour TWs given per supplier: $W_s = \{[1{:}00, 3{:}00], [3{:}00, 5{:}00]\}$, for $s \in \mathcal{S}$. From these sets, one TW is assigned to each supplier in the

**Figure 1** **Illustrative example of a network of suppliers and distribution centers with their specifications, plus the routes associated with two scenarios and the quantities of commodities transported on the links of the routes.**




first stage. The designated TW is denoted below each supplier triangle in Figure 1. Suppliers 1 and 2 are assigned the first TW, [1:00, 3:00]. Supplier 3 is assigned the second TW, [3:00, 5:00]. In the second-stage, 4 routes are selected from the pool of routes. Routes 1 and 2 are associated with scenario 1 and depicted in plain solid line and solid line with a plus sign, respectively. Routes 3 and 4 correspond to scenario 2 and are presented in plain dashed line and dashed line with a multiplication sign. There are two numbers reported above each link. The first number is the travel time (in minutes) between each two nodes and includes the service time, if the origin node is a supplier. The second number is either the star-service time at the destination node, if the destination is a supplier, or the arrival time to the destination node, if the destination is a distribution center. Below a link, on the other hand, the quantities of the commodities collected from suppliers are reported with the commodity IDs in parentheses. Note that the first link of a route departing from a distribution center does not carry any commodity and the final link of a route going back to the distribution center carries all the collected quantities from the suppliers. The capacity of the truck performing a route, meaning $Q$, is 25 units. Note that these routes are selected only for illustration purposes and the pool of routes is larger.

Truck routes visit the suppliers in their assigned TWs. From Figure 1, we see that depending on the earliness and lateness penalties, a decision maker might face a trade-off between assigning the first or second TW to supplier 3. This supplier is visited in both scenarios, on route 2 (in solid line with a plus sign) from scenario 1 and on route 3 (in dashed lines) from scenario 2. In the current solution, the second TW is assigned. As the TW-violation penalties are relatively high, the truck starts serving supplier 3 on route 2 (in solid line with a plus sign) at 3:00, which results in later arrival to distribution center 5. If the first TW, i.e. $[1{:}00, 3{:}00]$, was assigned to supplier 3, the start-service time at this supplier and the arrival time of the truck to distribution center 5 would have been (100 minutes) earlier on route 2. However, this TW-assignment would have also resulted in 10 minutes lateness in the start-service time of supplier 3 and consequently, an expensive penalty $(100 \times 10 = 1000)$ on route 3 (in dashed lines), as the start-service time at supplier 3 on that route is 3:10.

## 4. The Scenario Decomposition-Based Heuristic

In this section, we present our scenario decomposition-based heuristic, namely PHA. We first give a general overview and then describe the individual components in detail.

### 4.1. Overview

Setting TWs is challenging for a variety of reasons. First, demand may vary significantly. On some days, demand may be limited and only a few suppliers have to be visited. In this case, early TWs may be an advantage. On other days with higher demand, longer routes for visiting several suppliers need to be planned. Therefore, later TWs might be the right choice. Second, there is an interplay between the TW-decisions. Setting a TW for one supplier may change the sequence in which other suppliers are visited on a route. Thus, a TW-assignment decision has to consider the TWs of individual suppliers and the corresponding routing solutions holistically. Third, suppliers may be visited by vehicles from multiple distribution centers. Consequently, a supplier may be visited by vehicles from a close distribution center, which favors an early TW, but may also be visited from another center, which favors a later TW. Fourth, even if the demand was deterministic, the resulting multi-depot multi-commodity team orienteering problem with soft TWs would by itself be very challenging to solve.

To capture all these aspects, we apply a set of concerted steps. First, we consider a finite set of demand scenarios in which each scenario is one realization of the demand at the distribution centers. We then construct the deterministic equivalent problem (DEP), which is a multi-scenario approximation of the problem at hand. For the resulting DEP, we present a PHA-based solution approach that relies on a thorough consideration of possible routing solutions. PHA originates from linear programming (Rockafellar and Wets 1991). The idea of PHA is to first decompose (a relaxation of) DEP into subproblems associated with scenarios and then solve individual subproblems iteratively and force convergence by penalizing the deviation from the current consensus solution. We adapt the concept to our approach as follows. After decomposing DEP, our PHA-based heuristic iteratively solves deterministic scenario subproblems and determines a global TW-assignment per supplier, which might be infeasible for some scenarios. These global TW-assignments are used in the next iteration to guide the individual scenarios, and eventually, they will (hopefully) all provide the same solution. This solution is then implemented. While convergence is guaranteed for linear, and more generally convex, problems, it is not for mixed-integer problems. Here, right tuning and some enhancements are essential for successful application of PHA. For our problem, an additional challenge arises. That is solving the individual scenarios. Each individual scenario is a multi-depot multi-commodity team orienteering problem with soft TWs. To allow effective solutions for instances of realistic sizes, we model the problem as a route-based mixed-integer program and propose a matheuristic with a set of routing candidates capturing the variety of different demand scenarios for solving the subproblems. If after our PHA-based heuristic has run its course, there is no convergence and there are still some suppliers with no globally agreed TWs in the solution, similar to Crainic et al. (2011), we apply a repair phase to enforce consensus by selecting TWs that have been chosen the most among scenarios for those suppliers.

In the following, we first introduce the formulation of DEP in Section 4.2 for a finite number of scenarios and demonstrate the decomposition of DEP. We then present our PHA enhancements such as the construction of global TW-assignment based on the solutions of scenario subproblems in Section 4.3, the adjustments of PHA penalties in Section 4.4, and the termination of PHA and potential repairing of the final solution in Section 4.5. Section 4.6 illustrates our matheuristic for solving the individual subproblems. The section ends with an overview of the interplay of the individual steps in Section 4.7.

### 4.2. The Deterministic Equivalent Problem

If $\Omega \subseteq \bar{\Omega}$ is a finite set of scenarios for the random event, DEP is a multi-scenario deterministic approximation of our problem in which the second-stage objective function as well as constraints and variables ((4)–(16)) are defined for all scenarios, while the first-stage constraints and variables ((2)–(3)) stay unchanged. Note that in our underlying problem, there is no first-stage objective function. The probability of scenario $\omega \in \Omega$ is denoted by $p^\omega$. The formulation of DEP is as follows:

$$\min \sum_{\omega \in \Omega} p^\omega \Big\{ \sum_{s \in \mathcal{S}} \sum_{r \in R} \sigma_{sr} \left[ c_e e^\omega_{sr} + c_l l^\omega_{sr} \right] + \sum_{r \in R} \sum_{\substack{d \in \mathcal{D}: \\ \gamma(r)=d}} a^\omega_{dr} \Big\} \tag{17}$$

$$\text{s.t. Constraints (2)} - \text{(3)}.$$
$$\text{Constraints (5)} - \text{(16), for every}\, \omega \in \Omega.$$

Next, we decompose DEP into individual scenarios. We first modify DEP by creating copies of the first-stage variables as $x_{sw}^{\omega} \in \{0,1\}$, $s \in \mathcal{S}, w \in W_s$, per scenario $\omega \in \Omega$. This modification results in a formulation in which the objective function and all constraints and variables used in DEP are inherited unchanged, except constraints (2)–(3), (8), and (9). These constraints are transformed into:

$$\sum_{w \in W_s} x_{sw}^{\omega} = 1, \qquad s \in \mathcal{S}, \omega \in \omega \tag{18}$$

$$\sigma_{sr} e_{sr}^{\omega} \geq \underline{w} x_{sw}^{\omega} - \sigma_{sr} a_{sr}^{\omega} - M(1 - x_{sw}^{\omega}), \qquad s \in \mathcal{S}, w = [\underline{w}, \overline{w}] \in W_s, r \in \mathcal{R}, \omega \in \Omega \tag{19}$$

$$\sigma_{sr} l_{sr}^{\omega} \geq \sigma_{sr} a_{sr}^{\omega} - \overline{w} x_{sw}^{\omega} - M(1 - x_{sw}^{\omega}), \qquad s \in \mathcal{S}, w = [\underline{w}, \overline{w}] \in W_s, r \in \mathcal{R}, \omega \in \Omega \tag{20}$$

$$x_{sw}^{\omega} \in \{0,1\}, \qquad w \in W_s, s \in \mathcal{S}, \omega \in \Omega \tag{21}$$

respectively, plus a new set of constraints, i.e. “nonanticipativity constraints”;

$$x_{sw}^{\omega} = x_{sw}^{\omega'}, \qquad w \in W_s, s \in \mathcal{S}, \omega, \omega' \in \Omega, \omega \neq \omega' \tag{22}$$

are added. The goal of constraints (22) is to guarantee that TWs assigned to suppliers are scenario-independent. Then, in order to decompose the new formulation by scenarios, we replace constraints (22) with constraints

$$x_{sw}^{\omega} = \hat{x}_{sw}, \qquad w \in W_s, s \in \mathcal{S}, \omega \in \Omega \tag{23}$$

in which

$$\hat{x}_{sw} := \sum_{\omega' \in \Omega} p^{\omega'} x_{sw}^{\omega'}, \qquad w \in W_s, s \in \mathcal{S} \tag{24}$$

are global TWs assigned to suppliers and scenario-independent, but not necessarily feasible for every scenario (Rockafellar and Wets 1991). With this change, constraints (23) are then relaxed and added to the objective function via linear and quadratic penalty terms as followings:

$$\begin{aligned} \min \sum_{\omega \in \Omega} p^{\omega} \Big\{ & \sum_{s \in \mathcal{S}} \sum_{r \in R} \sigma_{sr} \left[ c_e e_{sr}^{\omega} + c_l l_{sr}^{\omega} \right] + \sum_{r \in R} \sum_{\substack{d \in \mathcal{D}: \\ \gamma(r) = d}} a_{dr}^{\omega} + \\ & \sum_{s \in \mathcal{S}} \sum_{w \in W_s} \lambda_{sw}^{\omega} (x_{sw}^{\omega} - \hat{x}_{sw}) + \frac{1}{2} \sum_{s \in \mathcal{S}} \sum_{w \in W_s} \rho (x_{sw}^{\omega} - \hat{x}_{sw})^2 \Big\} \end{aligned} \tag{25}$$

In (25), $\lambda_{sw}^{\omega}$ are penalty multipliers and $\rho$ is a quadratic penalty parameter associated with relaxed constraints (23), $w \in W_s, s \in \mathcal{S}, \omega \in \Omega$. Since TW-assignment variables $x_{sw}^{\omega}$, $w \in W_s, s \in \mathcal{S}, \omega \in \Omega$, are binary, similar to Crainic et al. (2011), we can reformulate objective function (25) in a linear format

$$\min \sum_{\omega\in\Omega} p^{\omega}\Big\{\sum_{s\in\mathcal{S}}\sum_{r\in R}\sigma_{sr}\left[c_e e^{\omega}_{sr}+c_l l^{\omega}_{sr}\right]+\sum_{r\in R}\sum_{\substack{d\in\mathcal{D}:\\ \gamma(r)=d}} a^{\omega}_{dr}+ \\ \sum_{s\in\mathcal{S}}\sum_{w\in W_s}\left[\lambda^{\omega}_{sw}-\rho\hat{x}_{sw}+\frac{\rho}{2}\right]x^{\omega}_{sw}-\sum_{s\in\mathcal{S}}\sum_{w\in W_s}\lambda^{\omega}_{sw}\hat{x}_{sw}+\sum_{s\in\mathcal{S}}\sum_{w\in W_s}\frac{\rho}{2}\hat{x}^2_{sw}\Big\}, \tag{26}$$

where the last two terms are constants. We apply the linearization, as quadratic mixed-integer problems are known to be more difficult to solve than linear mixed-integer ones (Veliz et al. 2015). Therefore, for a given global assignment of TWs of $\hat{x}_{sw}$, $w\in W_s, s\in\mathcal{S}$, the (linearized) relaxed formulation of the DEP can be decomposed by scenarios, which results in deterministic scenario subproblem SUB($\omega$), per scenario $\omega\in\Omega$, formulated as follows.

$$\min \sum_{s\in\mathcal{S}}\sum_{r\in R}\sigma_{sr}\left[c_e e^{\omega}_{sr}+c_l l^{\omega}_{sr}\right]+\sum_{r\in R}\sum_{\substack{d\in\mathcal{D}:\\ \gamma(r)=d}} a^{\omega}_{dr}+\sum_{s\in\mathcal{S}}\sum_{w\in W_s}\left[\lambda^{\omega}_{sw}-\rho\hat{x}_{sw}+\frac{\rho}{2}\right]x^{\omega}_{sw} \tag{27}$$

$$\text{s.t. Constraints } (5)-(7),(10)-(16),(18)-(21).$$

The penalty multipliers $\lambda^{\omega}_{sw}$ and penalty parameter $\rho$ serve the purpose of penalizing the deviation between the local TW-assignments in each SUB($\omega$) and the global TW-assignments $\hat{x}_{sw}$ for suppliers.

### 4.3. Constructing The Global Time Window Assignments

Since the first-stage variables, i.e. $x^{\omega}_{sw}$, in our underlying problem are binary variables, defining the global TW-assignments $\hat{x}_{sw}$ as in (24) will produce a $\{0,1\}$ value, if all scenarios agree on one TW per supplier. However, it might not always be the case and therefore, it can occur that $0<\hat{x}_{sw}<1$ for some suppliers $s\in\mathcal{S}$ and some TWs $w\in W_s$. Consequently, $\hat{x}_{sw}$ becomes an infeasible global TW-assignment for some suppliers. This phenomenon is quite probable in early iterations of PHA when the penalty multipliers and penalty parameter are not large. Over iterations, PHA plays with a trade-off between the penalty on deviating from the global TW-assignments, i.e. the current consensus TWs, and the cost of deviating from the locally TWs assigned in subproblems through earliness and lateness penalties in the objective function. Therefore, even though the values of $\hat{x}_{sw}$ might be infeasible in early iterations of PHA, they can still guide the search toward a consensus assignment of TWs for suppliers in the long run.

On the one hand, since our underlying problem is a team-orienteering problem, it is possible that some suppliers will not be visited in some (or all) individual subproblems. On the other hand, in every SUB($\omega$) a TW is assigned to each supplier in the first-stage via constraints (18), regardless of whether the supplier is visited on one of the routes chosen in the second-stage associated with scenario $\omega$. We believe that scenarios in which a supplier $s\in\mathcal{S}$ is not visited, i.e. no commodity is delivered, should not affect the search for consensus TWs. Therefore, in the calculation of $\hat{x}_{sw}$ in each iteration of PHA, we only take into account the values of first-stage variables from the scenarios where a supplier is visited and delivered a commodity on at least one route. To do so, we first define parameter $\chi(s,\omega)$ as the following:

$$\chi(s,\omega) = \begin{cases} 1, & \text{if} \, \exists r \in R, \exists m \in \mathcal{M}_s : \sigma_{sr} = 1, q^{\omega}_{msr} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

to check whether supplier $s \in \mathcal{S}$ is delivered something in scenario $\omega \in \Omega$. Then, we modify (24) to

$$\hat{x}_{sw} := \frac{\sum_{\substack{\omega\in\Omega:\\ \chi(s,\omega)=1}} p^{\omega} x^{\omega}_{sw}}{1 - \sum_{\substack{\omega\in\Omega:\\ \chi(s,\omega)=0}} p^{\omega}}. \qquad w \in W_s, s \in \mathcal{S} \tag{29}$$

In a special case where a supplier $s$ is not visited in any scenario, i.e. $\chi(s,\omega) = 0, \forall \omega \in \Omega$, we evaluate (29) by treating $\chi(s,\omega)$ as equal to one for each scenario. Another phenomenon that can happen over the iterations of PHA is that some values in $\hat{x}_{sw}$ can be very close for some TWs, for $s \in \mathcal{S}$. This means that there is no clear trend toward a single TW and there is a split among scenarios and the TWs they assign to suppliers locally. Such situations can result in no consensus achieved even after many iterations of PHA. Therefore, we apply a tie-breaking step on $\hat{x}_s$, $s \in \mathcal{S}$. We first start by checking

$$|\max_{w\in W_s}\{\hat{x}_{sw}\} - \min_{w\in W_s}\{\hat{x}_{sw}\}| \leq \eta, \tag{30}$$

for tie-breaking threshold $\eta$, which is a small enough positive value. If (30) is true, for $s \in \mathcal{S}$, we will then enforce consensus as

$$\begin{cases} \hat{x}_{sw^*} = 1, & \text{for} \, w^* = \arg\min_{w\in W_s}\{w\} \\ \hat{x}_{sw} = 0, & \text{for} \, w \in W_s : w \neq w^* \end{cases} \tag{31}$$

over the iterations of PHA. In other words, we choose the earliest TW as the tie-breaker for such suppliers.

### 4.4. Updating the Penalty Multipliers and Penalty Parameter

Rockafellar and Wets (1991)'s design of the original PHA is based on the augmented Lagrangean method for convex problems where the penalty multipliers associated with the nonanticipativity constraints are updated in each iteration. Assume $\lambda^{\omega,k}_{sw}$ is the penalty multiplier associated with nonanticipativity constraint (23) for supplier $s$ and TW $w$ in scenario $\omega$ from previous iteration $k$. The multiplier is then updated in the new iteration $k+1$ via:

$$\lambda^{\omega,k+1}_{sw} = \lambda^{\omega,k}_{sw} + \rho^k (x^{\omega,k+1}_{sw} - \hat{x}^{k+1}_{sw}), \tag{32}$$

with $x_{sw}^{\omega,k+1}$ being the solution of scenario subproblem SUB($\omega$) and $\hat{x}_{sw}^{k+1}$ being the global TW-assignment calculated by (29) in the new iteration, for $s \in \mathcal{S}, w \in W_s, \omega \in \Omega$. Moreover, Rockafellar and Wets (1991) suggest to update the penalty parameter iteratively as well. In the literature of PHA, the choice of the penalty parameter $\rho$ has been an important question, as the performance of the algorithm is highly sensitive to this value. Zehtabian and Bastin (2016) review different existing techniques and design an adaptive strategy based on the progress of PHA for updating $\rho$. Since our underlying problem is a mixed-integer linear programming problem, inspired by Crainic et al. (2011), we increase the value of $\rho$ dynamically and slowly. In addition, we impose a limit on the increase of the penalty parameter by introducing $\rho_{\max} > 0$. Let $\rho^k$ be the quadratic penalty parameter from previous iteration $k$. The penalty parameter is updated in the new iteration $k+1$ as:

$$\rho^{k+1} = \min(\mu\rho^k, \rho_{\max}), \tag{33}$$

where $\rho^0$ is an initial positive value for the penalty parameter and $\mu > 1$ is a constant step-size. Both values are chosen large enough to encourage convergence toward global TW-assignments but not too large such that the convergence is forced aggressively and prematurely.

### 4.5. Calculation of Consensus Deviation and Repairing the Time Windows

To verify how far the scenarios are from reaching consensus on global TW-assignments, we adapt the criterion $\sqrt{\sum_{\omega\in\Omega} p^\omega ||x^{\omega,k+1} - \hat{x}^k||_2^2} \le \varepsilon$, $x^{\omega,k+1}$ being the solution associated with scenario $\omega$ in the new iteration, $\hat{x}^k$ being the consensus vector calculated in the previous iteration of PHA, and $\varepsilon$ being the consensus threshold and a small enough value, from Zehtabian and Bastin (2016) to our underlying problem. Since we are in a team-orienteering context, similar to the calculation of $\hat{x}_{sw}$ in (29), we calculate the deviation from consensus TWs only for suppliers who are delivered a commodity at least on one route in a subproblem. In other words, we modify the criterion from Zehtabian and Bastin (2016) to

$$\sqrt{\sum_{\omega\in\Omega} \sum_{\substack{s\in\mathcal{S}: \\ \chi(s,\omega)=1}} \sum_{w\in W_s} p^\omega |x_{sw}^{\omega,k+1} - \hat{x}_{sw}^k|^2} \le \varepsilon, \tag{34}$$

with $\chi(s,\omega)$ defined in (28).

Convergence of PHA to a consensus solution is guaranteed, if the problem is convex (Rockafellar and Wets 1991). However, that is not the case, if the problem is non-convex (e.g. our underlying mixed-integer programming problem). So, it is possible that our PHA will run its course over iterations and not converge to a global assignment of TWs for all suppliers. If this happens, we apply a repair phase on the TW-assignments, as a an extra step. To that end, after final iteration of PHA, we verify whether there exists any supplier $s \in \mathcal{S}$ that still has non-binary values for some of the TWs, i.e. there is no consensus on TWs. If so, we repair the global TW-assignment for that supplier by applying the following rule:

$$\begin{cases} \hat{x}_{sw^{**}} = 1, & \text{for } w^{**} = \arg\max_{w \in W_s}\{\hat{x}_{sw}\} \\ \hat{x}_{sw} = 0, & \text{for } w \in W_s : w \neq w^{**} \end{cases} \tag{35}$$

In (35), we choose the TW with the biggest value $\hat{x}_{sw}$ among all $w \in W_s$, per supplier $s \in \mathcal{S}$. In case of multiple TWs with the same value, the earliest is chosen.

### 4.6. Solving the Individual Scenario Subproblems

In each iteration $k$ of our heuristic PHA, we solve a multi-depot multi-commodity team-orienteering problem SUB$(\omega, k)$ corresponding to scenario $\omega \in \Omega$ over a set of routes via a commercial solver. Route generation can be done by enumerating all routes, which becomes computationally expensive very fast even for small size instances (as shown in Section 6.1), or heuristically. Therefore, we design a heuristic that generate routes for each scenario demand and add them to the pool of routes, while making sure to avoid duplicates. To keep the size of the pool manageable for the model, we impose a maximum number of routes generated on our heuristic. Note that in our heuristic, a route is a sequence of suppliers to be visited and it starts and ends at the same distribution center. In order to satisfy the demand of a distribution center as much as possible when visiting a supplier and avoid generating suboptimal routes, we assume that there is an "*imaginary load*" per commodity on a route. For the purpose of the heuristic, we consider the three following conditions when inserting a supplier into a route: (a) a supplier is not put on a route from a distribution center, if it does not offer any of the commodities required by the distribution center; (b) a supplier is not inserted into a route from a distribution center, if the center has no unsatisfied demand left for the commodities that the supplier offers; and (c) the imaginary load on the route cannot exceed the truck capacity.

These conditions can be formulated as follows. Let $I_{curr}(r) = [I_{curr}(r,m)]_{m \in \mathcal{M}}$ be the vector of the current imaginary loads of commodities on route $r$ from distribution center $d$ with scenario demands $D^{\omega}_{dm}$, $m \in \mathcal{M}$. $I_{curr}(r,m)$ is the current imaginary load of commodity $m$ on route $r$ and initially set to 0. Note that $D^{\omega}_{dm} - I_{curr}(r,m)$ is the unsatisfied demand of distribution center $d$ for commodity $m$. Moreover, assume $\underline{Q}_{curr}(r)$ is the current remaining capacity on route $r$ and initially set to $Q$. To verify whether supplier $s$ can be put on route $r$, we calculate the potential new imaginary load for $m \in \mathcal{M}$ and the remaining capacity on route $r$ as

$$I_{new}(r,m) = I_{curr}(r,m) + \min\left(D^{\omega}_{dm} - I_{curr}(r,m), O_{sm}, \underline{Q}_{curr}(r)\right),$$
$$\underline{Q}_{new}(r) = \underline{Q}_{curr}(r) - \min\left(D^{\omega}_{dm} - I_{curr}(r,m), O_{sm}, \underline{Q}_{curr}(r)\right).$$

If $\underline{Q}_{new}(r) \neq \underline{Q}_{curr}(r)$, supplier $s$ can be put on route $r$ and we update $I_{curr}(r,m) \leftarrow I_{new}(r,m)$, $m \in \mathcal{M}$, and $\underline{Q}_{curr}(r) \leftarrow \underline{Q}_{new}(r)$. Our route generation scheme consists of two main independent phases. In phase 1, we first generate a set of routes through a restricted enumeration to ensure the feasibility of the MIP model. In phase 2, we expand the pool of routes in three steps by using a constructive heuristic and destroy and repair operators. The two phases are described as follows. The first phase contains a restricted enumeration of single-supplier and two-supplier routes from each distribution center. In both cases, we make sure that condition (a) mentioned above is enforced. In case of two-supplier routes, we put two suppliers on a route, if the distribution center is the closest center of both suppliers and conditions (b) and (c) are respected. These routes will then be added to the pool of routes, which is initially empty.

The second phase consists of three steps of generating a base set of routes via a constructive heuristic, post-processing those routes, and then constructing more routes by applying destroy and repair operators on the post-processed routes iteratively. For more details of these steps, we refer to Appendix A. To avoid having multiple copies of the same route added to the pool, we check whether a route generated in phase 1 or phase 2 already exists in the pool. If not, we then add the route to the pool. Moreover, in both phases we check the size of the pool before adding a route, to make sure not to violate the maximum number of routes generated.

### 4.7. All Steps Together

All steps of the procedure are put together in Algorithm 1. Since there is no guarantee that the scenarios will reach to a consensus on TWs assigned to suppliers, in addition to checking for consensus via equation (34), we consider a limit on the number of iterations in which PHA runs, set to 20, as another stopping criterion. If no consensus has been reached, we repair the global TW-assignments, as the last step.

Algorithm 1 starts by initializing the penalties of PHA $\lambda_{sw}^{\omega,0}$, $s \in \mathcal{S}, w \in W_s, \omega \in \Omega$, and $\rho^0$, computing global TW-assignments, i.e. $\hat{x}_{sw}^0$ per supplier $s$ and TW $w \in W_s$, based on the solutions of individual scenario subproblems with no penalty in the objective function (line 1). After the initialization step ($k=0$), as long as the stopping criteria are not met (lines 2-5), we compute a new local TW-assignment solution for each scenario by solving the corresponding subproblem with the penalty term back in the objective function (line 2). We then calculate a new global TW-assignment solution based on the local solutions of scenario subproblems and apply tie-breaking on $\hat{x}_s^{k+1}$, if necessary (line 3). The next step is to update the penalties $\lambda_{sw}^{\omega,k+1}$ and $\rho^{k+1}$ (line 4). If the limit on number of iterations is reached, but there are still some suppliers with no consensus over the global TWs, we repair the TW-assignments for suppliers with no consensus by choosing the TW with the biggest value among $\hat{x}_{sw}^{k+1}$.

**Algorithm 1:** The (heuristic) PHA

Initialization: Set $k \leftarrow 0$ and $\lambda_{sw}^{\omega,0} \leftarrow 0$, $s \in \mathcal{S}, w \in W_s, \omega \in \Omega$. Compute $\hat{x}_{sw}^{0} = \sum_{\substack{\omega\in\Omega:\\ \chi(s,\omega)=1}} p^{\omega} x_{sw}^{\omega,0} / 1 - \sum_{\substack{\omega\in\Omega:\\ \chi(s,\omega)=0}} p^{\omega}$, where $x_{sw}^{\omega,0}$ is the solution of scenario subproblem SUB$(\omega,0)$ without the penalty term in the objective function, for $\omega \in \Omega$. Choose $\rho^0 > 1$;

**repeat**

  Compute $x_{sw}^{\omega,k+1}$, for $s \in \mathcal{S}, w \in W_s$, by solving SUB$(\omega, k+1)$, for $\omega \in \Omega$

$$\min \sum_{s\in\mathcal{S}} \sum_{r\in R} \sigma_{sr} \left[ c_e e_{sr}^{\omega} + c_l l_{sr}^{\omega} \right] + \sum_{r\in R} \sum_{\substack{d\in\mathcal{D}:\\ \gamma(r)=d}} a_{dr}^{\omega} + \sum_{s\in\mathcal{S}} \sum_{w\in W_s} \left[ \lambda_{sw}^{\omega,k} - \rho^k \hat{x}_{sw}^{k} + \frac{\rho^k}{2} \right] x_{sw}^{\omega}$$

$$\text{s.t. Constraints } (5)-(7), (10)-(16), (18)-(21);$$

  Calculate new global TW-assignments $\hat{x}_{sw}^{k+1}$ by using equation (29) with $x_{sw}^{\omega,k+1}$ being the solution of SUB$(\omega, k+1)$, $\omega \in \Omega$. Apply tie-breaking (31) on $\hat{x}_{s}^{k+1}$, if (30) satisfies;

  Update penalties $\lambda_{sw}^{\omega,k+1}$, for $s \in \mathcal{S}, w \in W_s, \omega \in \Omega$, and $\rho^{k+1}$ via equations (32) and (33), respectively;

  Set $k \leftarrow k+1$;

**until** *The stopping criteria are met*;

If there are still suppliers with no consensus on TWs, repair the global TWs by using rule (35);

## 5. Design of Experiments

In this section, we describe the parameters of our heuristic PHA and the design of our numerical experiments. We further present the benchmark methods.

### 5.1. Parameters

Regarding the parameters for PHA, unless otherwise specified, we set the tie-breaking threshold $\eta$ to $10^{-1}$, the step-size $\mu$ for updating the penalty parameter to 1.25, the initial penalty parameter $\rho^0$ to $1 + \log(1 + N^0)$ (similar to Crainic et al. (2011)), where $N^0$ is the number of suppliers with no consensus TWs amongst the initial scenario solutions in the initialization step of PHA, and the consensus threshold $\varepsilon$ to $10^{-5}$. All numerical tests are implemented in C# and conducted on a machine with a 1.8 Gigahertz ADM Ryzen 7 5700U CPU and 16 GB of RAM. Unless otherwise stated, the scenario subproblems are solved heuristically by using a commercial solver, more specifically CPLEX 20.1. The solver stops if either the time limit of 30 minutes is reached or the CPLEX MIP gap is below 10%.

### 5.2. Instance Generation

We generate 10 instances each for small and large sizes, i.e., 20 instances overall. Both groups of instances are derived from the instances used in the study by Gu et al. (2021) in the context of a multi-commodity two-echelon distribution problem. Gu et al. (2021) have two main categories of instances: randomly generated instances and case study-based instances from a local fresh food supply chain. To generate our own instances, we select instances from their (reduced) base set of generated instances, denoted as $\mathscr{S}$ by Gu et al. (2021), and school canteens instances of the case study. The selected generated instances are used to construct our small instances and the selected school canteens instances are used to build our large instances. Although, we only have suppliers and distribution centers as stakeholders in our underlying problem, we use the customers from the selected instances per category from Gu et al. (2021) to generate demands for the distribution centers by projecting the customers' demands onto their closest distribution centers. In our small instances, we have five suppliers, two distribution centers, and supply and stochastic demand for three commodities, while in the large instances, we have 20 suppliers, five distribution centers, and supply and stochastic demand for eight commodities.

We generate three in-sample scenarios, as preliminary experiments showed that more scenarios increase complexity without a significant increase in the objective value. For the scenarios, we derive a base (average) demand per customer and commodity from the selected instances of Gu et al. (2021) and then generate a sample of three scenarios by perturbing these base demands with a random perturbation factor drawn from $[-0.25, 0.25]$ and multiplying the result by scenario-dependent factors drawn randomly from intervals $[0.60, 0.80]$, $[0.85, 1.05]$, and $[1.10, 1.30]$, per customer and commodity. We aggregate the demand of customers by their closest distribution centers, per commodity and use the aggregated demand as the demand of the distribution centers for the commodity. We also compute the available supply of commodities at the suppliers from the selected instances of Gu et al. (2021). In order to avoid demand-supply imbalance and infeasibility issues, we adjust the calculated supply according to the highest demand scenario plus 10% extra supply per supplier and commodity. Unless otherwise stated, the set of possible TWs consists of three one-hour TWs $[0, 60]$, $[60, 120]$, and $[120, 180]$ per supplier. The time windows capture the first half of the day, since in that time the collections from suppliers need to be done to ensure the deliveries to distribution centers in the second half. Each truck driver is assumed to work at most for six hours. Unless otherwise specified, the penalty on earliness, i.e. $c_e$, and on lateness, i.e. $c_l$, are both set to 10. For further details of our instance generation, we refer to Appendix B. Our generated instances are publicly available on GitHub (`https://github.com/szehtab/Consistent-Time-Window-Assignments`).

### 5.3. Benchmark Policies

In order to analyze the performance of our PHA, we propose five benchmark policies for assigning TWs: an optimal policy, an expected value policy, a multiple-scenario approach, our own multiple-scenario technique, and a priority rule.

1. We begin with solving DEP constructed based on the low, medium, and high demand scenarios to optimality. We do this only for our small instances due to computational intractability.

2. We then solve the expected value problem (EVP) for both small and large instances. EVP is a deterministic formulation of the two-stage stochastic programming formulation (1)-(16) in which the stochastic demands of the distribution centers are replaced by their expectations over the three scenarios for the centers.

3. We use a multiple-scenario approach (MSA, Bent and Van Hentenryck 2004) in which scenario subproblems are solved individually, without the penalty term in the objective. Then the scenario solutions are considered and the "most similar" one is selected. Thus, while PHA operates with average decision variables, MSA operates with "average" solutions. We determine similarity by the Hamming-distance metric on the TW-solution matrices following Song et al. (2020).

4. We also test a compromise between PHA and MSA. This technique, denoted by PHA(0), first solves the scenario subproblems without the penalty term in the objective, similar to the initialization step of PHA. Then, it constructs global TW-assignments via (29) based on scenarios' solutions and apply the repair mechanism (35), if there are suppliers with no consensus TWs.

5. Finally, we create a practically-inspired priority rule (P-Rule) which assigns TWs to suppliers based on the distance to their closest distribution centers. Closer suppliers get earlier TWs and more distant suppliers get later TWs. To this end, the suppliers are sorted in a list in an ascending order based on their distance to their closest distribution centers from the shortest distance to the longest distance. The first TW is then assigned to the first 33% of suppliers, the second TW to the next 33% of suppliers, and the third TW is assigned to the last portion of suppliers in that list. In its essence, P-Rule does not solve any form of the stochastic problem. However, since it is a simple and intuitively good enough policy to be implemented in practice, we design it for our numerical studies.

## 6. Numerical Experiments

In this section, we present the results of our numerical experiments. We first evaluate the solution quality of our PHA along with the benchmark policies. Moreover, we analyze the impact of offering more and smaller TWs to suppliers in the first stage on routing cost and TW-violations in the second stage. We finish by answering questions like what happens if the violations from TWs become cheaper or more expensive and whether there is a trade-off between the travel time and the penalty on TW-violation.

The results discussed in the first part of Section 6.1 from small instances are in-sample validation values over the set of the 3 base scenarios to assess the TW-assignments that are optimized on the same set of scenarios. The rest of the results reported in Section 6.1 and all other sections are out-of-sample values over a set of 25 additional scenarios that are sampled for the random event independently, per instance. For the details of out-of-sample validations, we refer to Appendix C.

**Table 3** Results of different policies–Small instances

| Instance | DEP | | MSA | | P-Rule | | PHA(0) | | PHA | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $z^{\text{in}}$ | CPU(h) | $z^{\text{in}}$ | Gap | $z^{\text{in}}$ | Gap | $z^{\text{in}}$ | Gap | $z^{\text{in}}$ | Gap |
| 1 | 626.1 | 3.3 | 741.9 | 18.5% | 644.6 | 3.0% | 673.4 | 7.5% | 651.8 | 4.1% |
| 2 | 580.4 | 1.7 | 604.9 | 4.2% | 640.6 | 10.4% | 604.9 | 4.2% | 585.6 | 0.9% |
| 3 | 652.6 | 0.3 | 817.8 | 25.3% | 692.4 | 6.1% | 652.6 | 0.0% | 720.4 | 10.4% |
| 4 | 630.1 | 7.6 | 677.1 | 7.5% | 634.1 | 0.6% | 677.1 | 7.5% | 648.1 | 2.9% |
| 5 | 524.1 | 0.1 | 577.8 | 10.3% | 533.6 | 1.8% | 524.1 | 0.0% | 540.3 | 3.1% |
| 6 | 693.4 | 5.3 | 791.7 | 14.2% | 835.4 | 20.5% | 791.7 | 14.2% | 694.3 | 0.1% |
| 7 | 729.8 | 0.5 | 796.1 | 9.1% | 729.8 | 0.0% | 729.8 | 0.0% | 729.8 | 0.0% |
| 8 | 586.8 | 3.5 | 622.7 | 6.1% | 627.1 | 6.9% | 622.7 | 6.1% | 586.8 | 0.0% |
| 9 | 617.2 | 11.3 | 663.8 | 7.6% | 626.0 | 1.4% | 624.9 | 1.3% | 622.3 | 0.8% |
| 10 | 611.2 | 3.0 | 662.2 | 8.3% | 614.6 | 0.6% | 611.2 | 0.0% | 611.2 | 0.0% |
| Average | - | - | - | 11.1% | - | 5.1% | - | 4.1% | - | 2.2% |

## 6.1. Solution Quality

In the following, we discuss the solution quality for the small and large instances.

**Small Instances.** We solve DEP to optimality for small instances over the base set of scenarios by enumerating all possible routes per instance. Next, we compare the objective values achieved from in-sample validation of the first-stage solutions (meaning the TW-assignments) from the PHA, PHA(0), MSA, and P-Rule policies to the true optimal from the DEP over the base demand scenarios. In order to find the optimality gap, we provide the fully enumerated set of routes to DEP as well as to PHA, PHA(0), MSA, and P-Rule for in-sample validation. Moreover, we turn off CPLEX solver settings for the time limit and MIP gap for in-sample validation of small instances. The results are presented in Table 3. There are two columns associated with every method. Columns $z^{\text{in}}$ and Gap (%) refer to the expected objective function value of in-sample validation with the first-stage solution of the corresponding policy and its optimality gap with the expected objective function value of DEP, respectively. CPU (in hours) points to the time it took DEP to return a solution. MSA (P-Rule) took (substantially) less than a second to get a first-stage solution. PHA(0) and PHA took no more than 4.5 seconds and six minutes to return a first-stage solution, respectively. The results in Table 3 show that PHA has the least gap, 2.2%, with DEP, on average. In six out of 10 instances, the PHA's gap is below 1% and there is only one outlier with a 10.4% gap. MSA followed by P-Rule and PHA(0) have the largest, second, and third largest gaps with DEP, on average. Notably is the poor performance of MSA which is outperformed even by the simple P-Rule policy. We conclude that PHA finds a first-stage solution that is almost equivalent to the one from DEP, with full set of routes on average, in much shorter time than DEP.

**Large Instances.** Solving DEP with a full enumeration of routes is not a viable benchmarking strategy for assessing the solution quality of the policies in large instances. Therefore, in order to analyze the first-stage solutions of the policies, i.e. TW-assignments, in large instances, we use the expected objective values obtained from out-of-sample validation and calculate the value of stochastic solution (VSS) per policy. VSS is generally used to show the advantage of solving a two-stage stochastic problem, in terms of expected cost saving, over solving EVP. For solving the two-stage stochastic problem, it is assumed that the stochastic information has a known probability distribution. Let $z^{\text{out}}_{EVP}$ and $z^{\text{out}}_{PHA}$ be the expected objective function values of the out-of-sample validations with the first-stage solutions of EVP and PHA, respectively. VSS is computed as VSS $= z^{\text{out}}_{EVP} - z^{\text{out}}_{PHA}$ and the relative VSS (R-VSS) is calculated

$$\text{R-VSS} = 100 \times \frac{z^{\text{out}}_{EVP} - z^{\text{out}}_{PHA}}{z^{\text{out}}_{EVP}}.$$

As our underlying problem is a minimization problem, the expected objective values of the EVP is considered as an upper bound for the objective value of the stochastic problem. Therefore, the expected objective values of out-of-sample validations with the first-stage solutions from different policies are expected to be smaller than the ones from the solution of EVP. Consequently, a positive R-VSS means that there is a value in solving the stochastic problem by a policy compared to a deterministic formulation of the problem, which makes decisions ignoring the uncertainty in the problem.

We summarize the results of out-of-sample validations with the first-stage solutions from PHA, PHA(0), MSA, P-Rule, and EVP policies in Table 4. There are three columns per policy, except for EVP and P-Rule. The numbers in column $z^{\text{out}}$ denote the expected objective function value of the validation of the corresponding policy. The second column provides the R-VSS per algorithm. Similar to Table 3, column CPU is the time (in hours) a policy took to return a solution. The solution of EVP is used for assessing the quality of the first-stage solutions of other policies by calculating the R-VSS. Hence, there are only $z^{\text{out}}$ and CPU columns associated with EVP in the table. P-Rule policy took maximum one second on average to get the first-stage solution, and so there are only $z^{\text{out}}$ and R-VSS columns presented for this algorithm in the table. From Table 4, we observe that the solution of PHA performs better than the deterministic solution of EVP, on average 2.5% in terms of the expected objective value for large instances. The solution of PHA outperforms the one from EVP per instance, ranging from 0% to 4% in terms of the expected objective value, as well. Although, the solution returned by PHA(0) also outperforms the solution of EVP, on average 1.7% in terms of expected objective value, its performance is volatile amongst the instances. Thus, while PHA may already produce some good (repaired) solutions after the initialization step, applying PHA over several iterations ensures better solutions throughout. The solution of the MSA performs on average 0.5% worse than the solution of EVP in terms of the expected objective value by having negative R-VSS values. Thus, for this setting, using MSA does

**Table 4 Results of different policies–Large instances**

| Instance | EVP | | P-Rule | | MSA | | | PHA(0) | | | PHA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $z^{\text{out}}$ | CPU(h) | $z^{\text{out}}$ | R-VSS | $z^{\text{out}}$ | R-VSS | CPU(h) | $z^{\text{out}}$ | R-VSS | CPU(h) | $z^{\text{out}}$ | R-VSS | CPU(h) |
| 1 | 1625.7 | 0.1 | 1667.6 | -2.5% | 1653.6 | -1.7% | 0.6 | 1612.4 | 0.8% | 0.6 | 1578.2 | 2.9% | 8.3 |
| 2 | 1542.3 | 0.4 | 1646.9 | -6.4% | 1550.3 | -0.5% | 1.0 | 1511.2 | 2.0% | 1.0 | 1494.8 | 3.1% | 13.5 |
| 3 | 1564.3 | 0.3 | 1634.5 | -4.3% | 1557.0 | 0.5% | 1.0 | 1510.8 | 3.4% | 0.9 | 1527.3 | 2.4% | 18.1 |
| 4 | 1589.2 | 0.1 | 1673.6 | -5.0% | 1651.8 | -3.8% | 0.6 | 1575.2 | 0.9% | 0.6 | 1525.3 | 4.0% | 7.9 |
| 5 | 1661.2 | 0.5 | 1759.1 | -5.6% | 1645.6 | 1.0% | 0.6 | 1620.8 | 2.4% | 0.6 | 1627.5 | 2.0% | 15.6 |
| 6 | 1820.4 | 0.3 | 1853.3 | -1.8% | 1919.1 | -5.1% | 0.8 | 1792.3 | 1.5% | 0.8 | 1791.1 | 1.6% | 13.4 |
| 7 | 1719.0 | 0.5 | 1728.2 | -0.5% | 1662.8 | 3.4% | 0.9 | 1649.8 | 4.0% | 0.9 | 1649.8 | 4.0% | 13.4 |
| 8 | 1667.7 | 0.1 | 1651.9 | 1.0% | 1629.2 | 2.4% | 0.6 | 1638.3 | 1.8% | 0.6 | 1627.6 | 2.4% | 15.4 |
| 9 | 1702.0 | 0.1 | 1781.8 | -4.5% | 1737.6 | -2.0% | 0.8 | 1739.7 | -2.2% | 0.8 | 1702.8 | 0.0% | 9.9 |
| 10 | 1666.8 | 0.2 | 1776.5 | -6.2% | 1654.9 | 0.7% | 0.1 | 1629.5 | 2.2% | 0.1 | 1628.4 | 2.3% | 4.5 |
| Average | - | - | - | -3.6% | - | -0.5% | - | - | 1.7% | - | - | 2.5% | - |

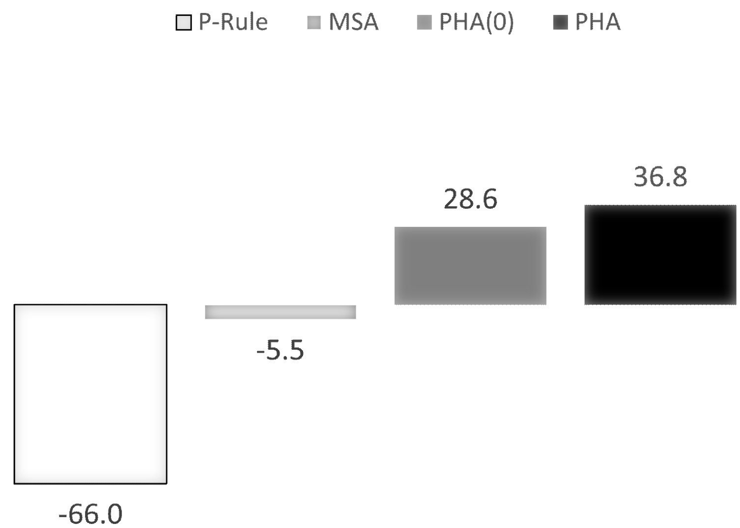
not lead to any improvement. Similarly, P-Rule not only underperforms on average 3.6%, but also in all instances but one falls behind EVP with regard to the expected objective value, as R-VSS is negative in nine out 10 instances.

To translate the value of the first-stage solution of each method, we present the expected arrival times of the routes (i.e., travel cost) to the distribution centers and the expected violation penalties averaged over 10 large instances per policy versus the first-stage solution of EVP in Figures 2a and 2b. The positive (negative) numbers in Figure 2a show the expected number of minutes routes in a policy's solution arrive earlier (later) to the distribution centers compared to the solution of EVP, on average. In other words, the positive (negative) values are the expected numbers of minutes saved (lost) by using the first-stage solution of any method compared to the one from EVP. All algorithms improve the arrival times to the distribution centers, i.e. save minutes, as to EVP, except P-Rule, on average. By comparing PHA and P-Rule, we also note that the trucks will not be busy for at least 100 minutes with the solution of PHA compared to the one from P-Rule on average. Therefore, by using PHA, a logistics company may expect on average more than 1.5 hours of paid labor by the drivers saved every day.

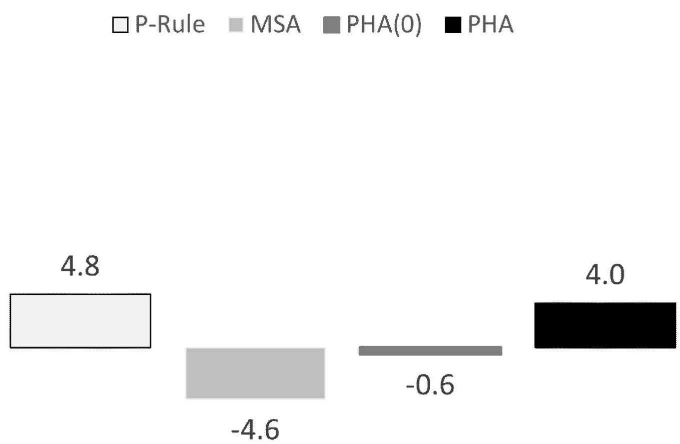
Figure 2b depicts the expected violation penalty of TWs in the solution of each policy compared to the solution of EVP. The positive (negative) numbers in the figure denote how much less (more) penalty the assigned TWs of a policy occur as to the ones by EVP, on average. P-Rule is the policy with the best record with regard to the expected violation penalty, i.e. with the least expected violation penalty on average. However, regarding the expected arrival times, P-Rule has the worst performance, meaning it loses the most expected number of minutes. This shows that even though assigning TWs to the suppliers in a naive way and only based on the proximity might cause the least TW-violations, it results in the longest arrival times to the distribution centers (or biggest loss of minutes), as it completely ignores the presence of stochasticity in the problem and the routing part of the planning. Unlike P-Rule, MSA, in spite of having a negative R-VSS, performs

**Figure 2 Quality assessment of four policies via the expected objective terms–Large instances.**

(a) Expected number of minutes the solution of a policy saves (positive values) or loses (negative values) in the arrival times of routes to the distribution centers compared to the solution of EVP.




(b) Expected violation penalty on the routes that the solution of a policy causes less (positive values) or more (negative values) than the solution of EVP.




better than EVP regarding the expected arrival times on average. However, the expected violation penalty caused by the first-stage solution of MSA is the largest violation among the four policies compared to the one from EVP, hence R-VSS becomes negative. The underperformance of MSA is to the contrary of the prior knowledge of the performance of this policy in the literature (e.g. Song et al. 2020). Although, the first-stage solution of PHA(0) saves minutes in the expected arrival times to the distribution centers, compared to the solution of EVP, it results in more expected violation penalty on average. Overall, PHA has the best performance regarding both the expected arrival times and TW-violations compared to EVP among the four policies, on average.

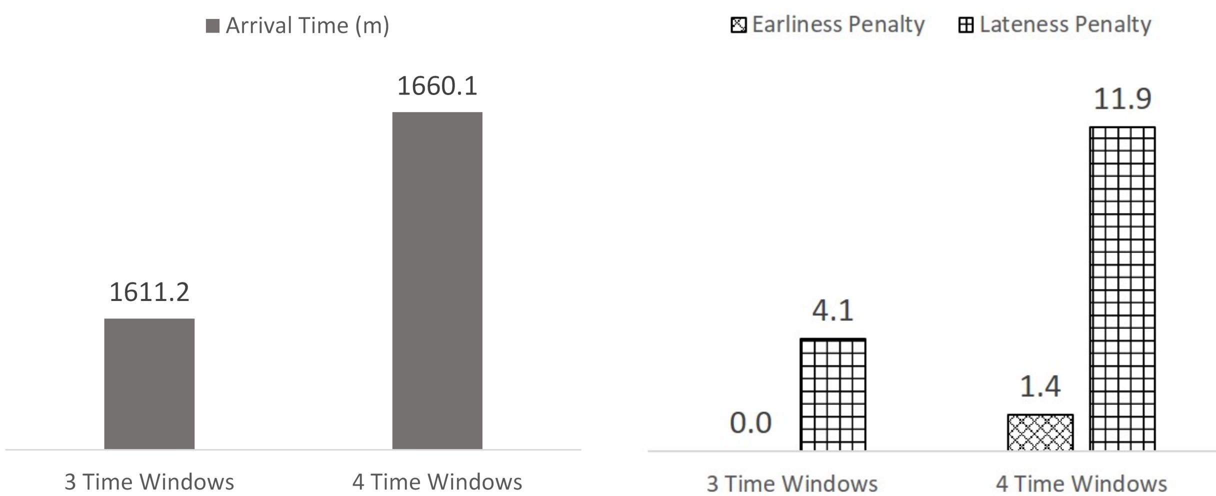
In conclusion, PHA and PHA(0), with 1.7% and 2.5% R-VSS respectively, have similarly good performance compared to EVP, among the four policies. While, PHA takes longer to find a first-stage solution than PHA(0), its solution provides more value than the one from PHA(0). Moreover, the arrival times of the routes from the solutions of these two policies are the earliest among the four policies compared to EVP. However, TWs assigned in the solution of PHA result in less violation penalty than the ones from EVP, while the TW-assignments from PHA(0) cause more violation penalty than TWs of EVP. Therefore, a decision maker has to deal with a trade-off regarding which first-stage solution to use.

### 6.2. Time Windows

There is a managerial trade-off between the design of TWs and the difficulty level of routing. Suppliers will probably like fine-grained and shorter TWs. However, this increases the complexity for the logistics provider in finding efficient routes that meet these narrow TWs. In order to analyze the consequences of offering more TWs on the routing cost, i.e. arrival times of routes to the distribution centers, and cost of violating the assigned TWs, we test the idea of offering four TWs in the large instances. These four TWs cover the same time duration as the initial three TWs, i.e. three hours time span, but they are shorter, meaning each is a 45 minutes interval, instead of 60 minutes. We use the PHA policy to find a first-stage solution for our underlying problem with this new change and as previously mentioned, we validate the solution on the set of the 25 validation scenarios, per instance. We then compare the expected arrival times of the routes to the distribution centers (in minutes) and the expected earliness and lateness penalties averaged over the 10 large instances with four TWs versus the similarly calculated numbers with the four TWs.

The results are presented in Figure 3. We observe that by offering one extra TW, i.e. offering more flexibility, all terms of the objective function increase on average. Compared to the case of having three TWs to choose from, the expected arrival times of routes to the distribution centers in case of four TWs rise by 3%, which can be translated into trucks losing more time by 3% on average. The expected violations from the assigned TWs also increase in case of offering four TWs, with a 136% and 188% jump in the earliness and lateness costs respectively, on average. Note that the earliness and lateness penalties are set to the same value in the objective function for both three and four TWs options. From these numbers, we can conclude that although offering one extra TW can provide more flexibility for the suppliers, it will add more complexity to the routing decisions in the second stage and possibly create more inconvenience for the suppliers, as the earliness and lateness costs suggest.

**Figure 3** **Comparison between three vs. four TWs through the expected arrival times to distribution centers (in minutes), earliness, and lateness penalties–Large instances.**




### 6.3. Severity of Time Window Violation

From the managerial insight perspective, if the TW-violation penalty is low, it will be easier to plan the routes and the logistics cost will drop, but the suppliers will not be served on a consistent schedule. On the other hand, if the violation penalty is high, it will be more difficult to plan the routes and the logistics cost will rise. However, the suppliers will be more likely to be visited according to the schedule and hence, they will be happy. To study the effect of the penalty of violating TWs on routing cost, which manifests itself in the arrival times of the routes to the distribution centers and TW-violations, we vary the value of the violation penalty from its initial value of 10 to the values of zero, one, and 100 for the large instances and use the PHA policy to find a first-stage solution. We then validate the solutions of these tests on the 25 validation scenarios for each instance. Finally, we compare the expected arrival times of the routes to the distribution centers (in minutes) and the expected number of minutes violating TWs, i.e. earliness and lateness, averaged over the 10 large instances for the four penalty values. The results are presented in Figure 4.

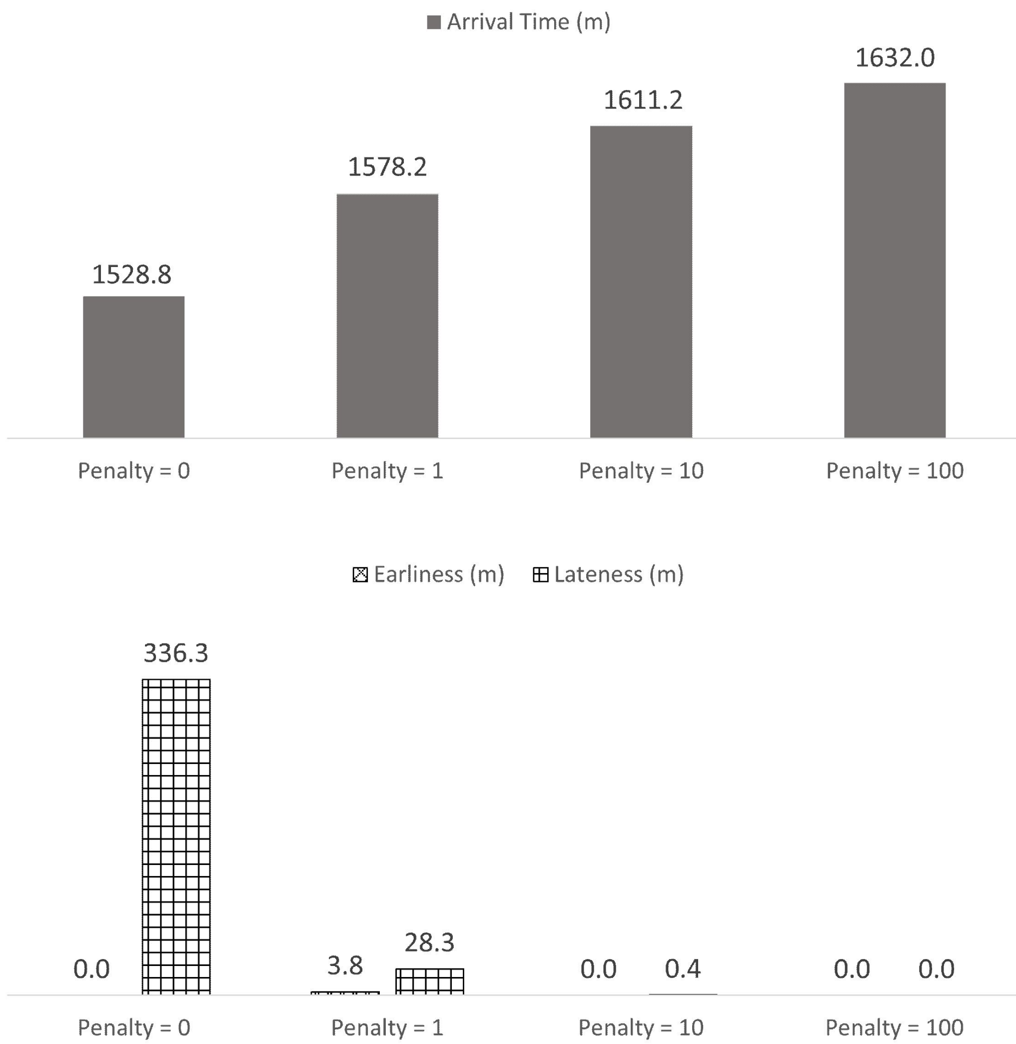
The upper chart in Figure 4 displays the expected arrival times of the routes to the distribution centers. We see that they increase (decrease), the more (less) critical TWs become. When the penalty on violating TWs is zero, technically the assigned TWs become ineffective and the only mechanism that affects planning of the routes is to have the arrival times of the trucks to distribution centers as early as possible. Therefore, no penalty case has the smallest expected arrival times (in minutes) to the centers, meaning the trucks save the most number of minutes, on average. When the penalty on violating TWs is set to 100, TWs become hard and the routes must be planned in such a way that there is no earliness or lateness in visiting the suppliers. This will result in delaying the expected arrival times of routes to the distribution centers, meaning the trucks will lose more minutes, by 6.8% on average. For the intermediate penalties of one and 10, the expected arrival times increase by 3.2% and 2.1%, compared to the case with penalty of zero and the case with penalty of one respectively, on average. The lower chart in Figure 4, on the other hand, shows the expected number of minutes violating the assigned TWs on average. Violation penalty value of zero has the biggest violation from TWs, as technically TWs are deactivated. The stronger TWs are enforced, i.e. violation penalties increase, the less TW-violations happen. When the violation penalty is at its maximum value of 100, TWs are strict and hence, no TW-violation occurs. Violation penalty value of one activates TWs, but since the violation cost is not high, there are both earliness and lateness. Overall, we conclude that having strict TWs can increase the busy time of the trucks by almost 6.8%, and at the same it makes the service of suppliers more consistent. Therefore, a decision maker should keep this price in mind when promising TWs to the suppliers.

## 7. Outlook

In this paper, we have introduced the problem of assigning consistent TWs for the collection of regional groceries from local farmers and delivering them to distribution centers for consolidation and further distribution in a short agri-food supply chain with stochastic demand. We have shown how considering demand uncertainty via the progressive hedging algorithm can reduce the routing cost as well as inconvenience for the suppliers. There are a variety of avenues for future research in problem and methodology.

In our problem, we assumed consistency was achieved when visits took place within TW. However, given the multi-depot setting of our problem, this means that a supplier might be visited by several vehicles at any point within TWs. Future research may investigate the cost of synchronization among the depots to ensure even less inconvenience for the suppliers. We have also shown that narrow TWs increase cost and violations significantly. Future research may consider heterogeneous TW-sizes for different suppliers, e.g., based on their usual position in a route. Another interesting problem extension could be the consideration of uncertain service times dependent on the commodity or even the supplier. Thus, the second stage problem would turn into a stochastic problem itself. From a methodology perspective, we have shown that our progressive hedging approach provides superior solutions compared to a multiple-scenario approach. As both approaches focus on different aspects of the scenario solutions, future research may have a more detailed comparison of the two types of methodology. Furthermore, future research may investigate combinations, for example, by integrating the multi-scenario approach in the determination of the global solution

**Figure 4** **Evolution of the expected arrival times of routes to distribution centers and earliness and lateness minutes for varying violation penalty–Large instances.**




within each iteration. Finally, in this work, we have focused on agri-food supply chains. However, the challenge of TW-consistency with multiple vehicles and goods can also be observed in other domains, e.g., in the delivery to stores and supermarkets. Future research may adapt model and methodology to these related settings.

## Acknowledgments

Marlin Ulmer's work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Emmy Noether Programme, project 444657906. We gratefully acknowledge their support.

## References

Ackva C, Ulmer M, 2022 *Consistent routing for local same-day delivery via micro-hubs.* Technical Report 7/2022, FEMM Working Paper Series, Magdeburg, Germany.

Archetti C, Campbell AM, Speranza MG, 2016 *Multicommodity vs. single-commodity routing. Transportation Science* 50(2):461–472.

Archetti C, Speranza MG, 2012 *Vehicle routing problems with split deliveries. International Transactions in Operational Research* 19(1-2):3–22.

Bent RW, Van Hentenryck P, 2004 *Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research* 52(6):977–987.

Birge JR, Louveaux F, 2011 *Introduction to Stochastic Programming.* Springer Series in Operations Research and Financial Engineering (USA: Springer New York, NY), 2nd edition.

Bosona TG, Gebresenbet G, 2011 *Cluster building and logistics network integration of local food supply chain. Biosystems Engineering* 108(4):293–302.

Campelo P, Neves-Moreira F, Amorim P, Almada-Lobo B, 2019 *Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. European Journal of Operational Research* 273(1):131–145.

Carlsson JG, Delage E, 2013 *Robust partitioning for stochastic multivehicle routing. Operations Research* 61(3):727–744.

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

Crainic T, Errico F, Malucelli F, Nonato M, 2012 *Designing the master schedule for demand-adaptive transit systems. Annals of Operations Research* 194(1):151–166.

Crainic TG, Fu X, Gendreau M, Rei W, Wallace SW, 2011 *Progressive hedging-based meta-heuristics for stochastic network design. Networks* 58(2):114–124.

Dalmeijer K, Desaulniers G, 2021 *Addressing orientation symmetry in the time window assignment vehicle routing problem. INFORMS Journal on Computing* 33(2):495–510.

Dalmeijer K, Spliet R, 2018 *A branch-and-cut algorithm for the time window assignment vehicle routing problem. Computers & Operations Research* 89:140–152.

Gu W, Archetti C, Cattaruzza D, Ogier M, Semet F, Speranza MG, 2021 *A sequential approach for a multi-commodity two-echelon distribution problem. Computers & Industrial Engineering* 107793.

Haughton M, 2007 *Assigning delivery routes to drivers under variable customer demands. Transportation Research Part E: Logistics and Transportation Review* 43(2):157–172.

Haugland D, Ho SC, Laporte G, 2007 *Designing delivery districts for the vehicle routing problem with stochastic demands. European Journal of Operational Research* 180(3):997–1010.

Jabali O, Leus R, Van Woensel T, de Kok T, 2015 *Self-imposed time windows in vehicle routing problems. OR Spectrum* 37(2):331–352.

Jalilvand M, Bashiri M, Nikzad E, 2021 *An effective progressive hedging algorithm for the two-layers time window assignment vehicle routing problem in a stochastic environment. Expert Systems with Applications* 165, 113877.

Kovacs A, Golden B, Hartl R, Parragh S, 2014 *Vehicle routing problems in which consistency considerations are important: A survey. Networks* 64(3):192–213.

Martins S, Ostermeier M, Amorim P, Hëbner A, Almada-Lobo B, 2019 *Product-oriented time window assignment for a multi-compartment vehicle routing problem. European Journal of Operational Research* 276(3):893–909.

Montoya-Torres JR, López Franco J, Nieto Isaza S, Felizzola Jiménez H, Herazo-Padilla N, 2015 *A literature review on the vehicle routing problem with multiple depots. Computers & Industrial Engineering* 79:115–129.

Mor A, Speranza MG, 2022 *Vehicle routing problems over time: A survey. Annals of Operations Research* 314:255—275.

Neves-Moreira F, Pereira da Silva D, Guimarães L, Amorim P, Almada-Lobo B, 2018 *The time window assignment vehicle routing problem with product dependent deliveries. Transportation Research Part E: Logistics and Transportation Review* 116:163–183.

Palacios-Argüello L, Sanchez-Diaz I, Gonzalez-Feliu J, Gondran N, 2020 *The role of food hubs in enabling local sourcing for school canteens.* Aktas E, Bourlakis M, eds., *Food Supply Chains in Cities: Modern Tools for Circularity and Sustainability*, 233–263 (Switzerland: Springer: Palgrave Macmillan Cham).

Palomo-Martínez PJ, Salazar-Aguilar MA, Laporte G, 2017 *Planning a selective delivery schedule through adaptive large neighborhood search. Computers & Industrial Engineering* 112:368–378.

Prajapati D, T S Chan F, Daultani Y, Pratap S, 2022 *Sustainable vehicle routing of agro-food grains in the e-commerce industry. International Journal of Production Research* 60(24):7319–7344.

Rockafellar RT, Wets RJB, 1991 *Scenario and policy aggregation in optimization under uncertainty. Mathematics of Operations Research* 16(1):119–147.

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

Schmidt-Forth A, 2022 *Saisonal und regional: So arbeiten Kantinen bio und klimafreundlich.* `https://www.augsburger-allgemeine.de/geld-leben/augsburg-saisonal-und-regional-so-arbeiten-kantinen-bio-und-klimafreundlich-id63701981.html`, accessed: 31-01-2023.

Solomon MM, 1987 *Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research* 35(2):254–265.

Song Y, Ulmer M, Thomas B, Wallace S, 2020 *Building trust in home services—Stochastic team-orienteering with consistency constraints. Transportation Science* 54(3):823–838.

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

Spliet R, Dekker R, 2016 *The driver assignment vehicle routing problem. Networks* 68(3):212–223.

Spliet R, Desaulniers G, 2015 *The discrete time window assignment vehicle routing problem. European Journal of Operational Research* 244(2):379–391.

Spliet R, Gabor AF, 2015 *The time window assignment vehicle routing problem. Transportation Science* 49(4):721–731.

Stavropoulou F, Repoussis PP, Tarantilis CD, 2019 *The vehicle routing problem with profits and consistency constraints. European Journal of Operational Research* 274(1):340–356.

Subramanyam A, Gounaris CE, 2018 *A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. Transportation Science* 52(2):386–401.

Subramanyam A, Wang A, Gounaris CE, 2018 *A scenario decomposition algorithm for strategic time window assignment vehicle routing problems. Transportation Research Part B: Methodological* 117:296–317.

Sungur I, Ren Y, Ordóñez F, Dessouky M, Zhong H, 2010 *A model and algorithm for the courier delivery problem with uncertainty. Transportation Science* 44(2):193–205.

Truchot D, Andela M, 2018 *Burnout and hopelessness among farmers: The farmers stressors inventory. Social Psychiatry and Psychiatric Epidemiology* 53(8):859–867.

USDA, 2022 *USDA expands local foods in school meals through cooperative agreement with minnesota.* `https://www.usda.gov/media/press-releases/2022/08/23/usda-expands-local-foods-school-meals-through-cooperative-agreement`, accessed: 31-01-2023.

Vareias A, Repoussis P, Tarantilis C, 2019 *Assessing customer service reliability in route planning with self-imposed time windows and stochastic travel times. Transportation Science* 53(1):256–281.

Veliz FB, Watson JP, Weintraub A, Wets RJB, Woodruff DL, 2015 *Stochastic optimization models in forest planning: A progressive hedging solution approach. Annals of Operations Research* 232:259–274.

Yadav VS, Singh AR, Gunasekaran A, Raut RD, Narkhede BE, 2022 *A systematic literature review of the agro-food supply chain: Challenges, network design, and performance measurement perspectives. Sustainable Production and Consumption* 29:685–704.

Zehtabian S, Bastin F, 2016 *Penalty parameter strategies in progressive hedging algorithm.* Technical Report 2016–12, CIRRELT, Montreal, Qc, Canada.

Zhong H, Hall RW, Dessouky M, 2007 *Territory planning and vehicle dispatching with driver learning. Transportation Science* 41(1):74–89.

## Appendix A: Steps of the Second Phase of Route Generation

Step1. Generating a base set of routes via a constructive heuristic: from each distribution center, we construct an initial set of routes by using the Savings algorithm from Clarke and Wright (1964), while making sure that conditions (a), (b), and (c) mentioned above are respected. We then improve these routes by applying a local search, namely a series of intra-route 2-opt moves, on them. The routes are then added to the pool of routes which already contains routes from the first phase.

Step2. Post-processing the base set of routes: the routes in Step1 are constructed from every distribution center and therefore, suppliers can appear on multiple routes. To avoid operating on copies of the same supplier, we deduplicate the suppliers. In other words, we keep a supplier on the route from a randomly chosen distribution center and remove it from the routes of other distribution centers.

Step3. Constructing more routes by applying destroy and repair operators iteratively: borrowed from the Large Neighborhood Search algorithm, we apply destroy and repair moves on the deduplicated routes iteratively to produce more routes that are diverse as well. In each iteration, we use two operators presented in Ropke and Pisinger (2006): the random removal and basic greedy insertion with a random noise on the insertion cost. We randomly remove a certain number of suppliers (a random number between 20% and 50% of the suppliers) and then insert back the removed suppliers into the routes by applying the basic greedy insertion with a random noise, while making sure that conditions (a), (b), and (c) mentioned above are respected by the insertion. Similar to Ropke and Pisinger (2006), we calculate the noise by drawing a number from interval $[-\max N, \max N]$ randomly, where $\max N = \eta \max_{j,j' \in \mathcal{V}}\{t_{jj'}\}$ and $\eta = 0.025$. At each iteration, we calculate the noise and add it to the insertion cost for deciding on which route to insert a supplier. Note that there is a possibility that one or more suppliers cannot be inserted back into any route due to conditions (a), (b), and (c). If this happens, i.e. there are some suppliers unserved on any existing route, we create empty routes from the distribution centers and start inserting the unserved suppliers into those routes via the basic greedy insertion operator with random noise. For this second insertion, we again ensure that conditions (a), (b), and (c) are respected. At the end of each iteration of destroy and repair, we apply a series of intra-route 2-opt moves on the routes before adding them to the pool. We apply destroy and repair operators for 100 iterations.

## Appendix B: Details of Instance Generation

Let $\mathcal{C}$ be the set of customers. These customers have the same locations but different demands in the selected instances of Gu et al. (2021). We compute the mean demand per $(i,m)$ pair, for customer $i \in \mathcal{C}$ and commodity $m \in \mathcal{M}$, and use it as the base demand, denoted by $\widetilde{D}_{im}$, for each pair. These base demands are then used to generate demand scenarios in our small and large instances. For each instance in their corresponding instance group, we generate a sample of three demand scenarios; low, medium, and high demand, with (almost) equal probabilities. This way, we can replicate the customers' demand behaviour, specially in case of school canteens. If the demand for fresh products increases or decreases over a planning horizon, it is quite possible that the demand for those products in other school canteens will follow the same trend. The demand per scenario $\omega$ is calculated as $D^{\omega}_{im} = \lceil \widetilde{v}^{\omega}_{im}(1+\widetilde{\nu}_{im})\widetilde{D}_{im} \rceil$ in which $\widetilde{D}_{im}$ is the base demand, $\widetilde{\nu}_{im}$ is a uniformly distributed perturbation factor on $[-0.25, 0.25]$, and $\widetilde{v}^{\omega}_{im}$ is a scenario-dependent uniformly distributed multiplier on intervals $[0.60, 0.80]$, $[0.85, 1.05]$, and $[1.10, 1.30]$ for $\omega \in \{1, 2, 3\}$, i.e. for low, medium, and high demands, respectively for $i \in \mathcal{C}, m \in \mathcal{M}$. We then project the customers' demand on their closest distribution centers to create demand scenarios for the distribution centers in each instance. In other words, we compute $D^{\omega}_{dm} = \sum_{\substack{i \in \mathcal{C}: \\ \gamma(i)=d}} D^{\omega}_{im}$, for $d \in \mathcal{D}$ and $\omega \in \{1, 2, 3\}$, where $\gamma(i)$ is defined as the closest distribution center to customer $i \in \mathcal{C}$. This sample of three scenarios is then used to obtain a solution per instance. As the available supply at suppliers is the deterministic part of our problem, next we describe a procedure for generating a base supply in each instance set which will then be used to generate supplies for small and large instances, accordingly.

The suppliers in the selected instances of Gu et al. (2021) have the same locations but different supply. To have a base supply of commodities for our instance generation, we set each supplier's available quantity of a commodity to its maximum among the selected instances. These base supplies are then adjusted with respect to the high demand scenario plus a 10% supply flexibility to make sure that no matter the demand scenario, there is enough supply available per commodity. Mathematically speaking, for each of our generated instances $O_{sm}$ is computed as $O_{sm} \leftarrow \lceil 1.10 \frac{\sum_{d \in \mathcal{D}} D^{\omega}_{dm}}{\sum_{s \in \mathcal{S}} \widetilde{O}_{sm}} \widetilde{O}_{sm} \rceil$, where $\widetilde{O}_{sm}$ is the base supply derived from the selected instances of Gu et al. (2021) and $\omega = 3$, i.e. the high demand scenario, for $m \in \mathcal{M}$ and $s \in \mathcal{S}$.

The travel time between each two locations is computed as the travel distance between the two locations multiplied by an instance-dependent factor such that the travel time between the two farthest locations in the instance is 90 minutes, plus 30 minutes service time if the first location of the two is a supplier. The activation cost of a truck (in minutes), i.e. $\tau$, is set to 60. We set the truck capacity to four times of the maximum capacity among the selected instances. The reason behind multiplying the maximum capacity by four is that we tested different values and four was the value that allowed the collection of commodities from more than two suppliers to satisfy the demand of the distribution centers in our numerical studies.

To generate our small instances, we take instances from the (reduced) base set of generated instances in Gu et al. (2021), denoted as $\mathscr{S}$ by the authors. We choose three with the largest demand. They are composed of six suppliers, two distribution centers, and 25 customers with locations produced based on the coordinates in the class $C101$ of the well-known Solomon (1987) set of instances. The number of commodities is either two or three. The travel distance between two locations is the Euclidean distance between the two nodes. The capacity of a truck collecting the supply is either 150, 180, or 209 units. We diversify the locations of suppliers by randomly choosing two suppliers out of 6 and exchanging their coordinates with the coordinates of the distribution centers for each of our generated instances. We reduce the number of suppliers to five by randomly removing one of the suppliers for each of our generated instances. We then generate customers' demand scenarios and project them on distribution centers following the procedure explained above. We set the number of commodities to three. The truck capacity is $4 \times 209 = 836$ units.

To build our large instances, we use the school canteens instances from the case study described in Gu et al. (2021). The case study was conducted by the local authorities of the French department of Isère. We choose the first six instances which all have 103 customers, five distribution centers, and 61 suppliers. The number of commodities, i.e. fresh products, varies from five to eight. The distance matrix between the locations are provided by the local authorities. The capacity of truck collecting the supply at the farmers is

600 units in all of these instances. We reduce the number of suppliers to 20 by deleting 21 suppliers randomly and projecting half of the remaining suppliers onto their closest neighboring supplier for each of our large instances. We then create customers' demand scenarios and project them on distribution centers following the steps mentioned above. We set the number of commodities to eight. The capacity of the trucks are set to $4 \times 600 = 2400$.

## Appendix C: Details of Out-of-sample Validation

The out-of-sample scenarios are used to validate the TWs optimized for the three base, i.e. low, medium, and high, demand scenarios generated according to the procedure described in Section 5.2. We refer to the set of these three scenarios used for optimization as the base set and the set of 25 scenarios used for out-of-sample validation as the validation set. The calculation of available supplies $O_{sm}$, $s \in \mathcal{S}$ and $m \in \mathcal{M}$, are the same as explained in Section 5.2. We then build 25 demand scenarios for customers by first calculating $\bar{D}^{\omega'}_{im} = \lceil \bar{v}^{\omega'}(1 + \bar{\nu}^{\omega'}_{im})(1 + \widetilde{\nu}_{im})\widetilde{D}_{im} \rceil$ for $i \in \mathcal{C}, m \in \mathcal{M}$, in which $\widetilde{D}_{im}$ is the base demand, $\widetilde{\nu}_{im}$ and $\bar{\nu}^{\omega'}_{im}$ are uniformly distributed perturbation factors on $[-0.25, 0.25]$, and $v^{\omega'}$ is a scenario-dependent uniformly distributed multiplier on interval $[0.40, 1.50]$, for $\omega' \in \{1, 2, \ldots, 25\}$. To make sure that these 25 validation scenarios are different than the three base scenarios, we use two different seeds for the two sets of scenarios. To avoid infeasibility due to imbalance supply and demand, if $\sum_{i \in \mathcal{C}} \bar{D}^{\omega'}_{im} > \sum_{s \in \mathcal{S}} O_{sm}$ for $m \in \mathcal{M}$, we then adjust the generated demand per customer $i \in \mathcal{C}$ and commodity $m \in \mathcal{M}$ via $D^{\omega'}_{im} \leftarrow \lfloor \frac{\sum_{s \in \mathcal{S}} O_{sm}}{\sum_{i \in \mathcal{C}} \bar{D}^{\omega'}_{im}} \bar{D}^{\omega'}_{im} \rfloor$, $\omega' \in \{1, 2, \ldots, 25\}$. Finally, similar to Section 5.2, we aggregate the demand of customers based on a shared closest distribution center and project the result as the demand of the center, i.e. $D^{\omega'}_{dm} = \sum_{\substack{i \in \mathcal{C}: \\ \gamma(i) = d}} \bar{D}^{\omega'}_{im}$, for $d \in \mathcal{D}$ and $\omega' \in \{1, 2, \ldots, 25\}$. The heuristic pool of routes contains routes that are generated by taking into account both the base and validation sets of scenarios.

**Otto von Guericke University Magdeburg**
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

**www.fww.ovgu.de/femm**

ISSN 1615-4274