KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

# Optimizing the workforce schedule and collection routes for glass containers

De Bruecker P, De Jaeger S, Beliën J,
Demeulemeester E, De Boeck L, Van den Bergh J.

KBI_1527

# Optimizing the Workforce Schedule and Collection Routes for Glass Containers


Philippe De Bruecker*

*KU Leuven, Research Center for Operations Management, Leuven (Belgium)*

Simon De Jaeger

*KU Leuven, Center for Information Management, Modeling and Simulation (CIMS), Brussels (Belgium)*

Jeroen Beliën

*KU Leuven, Center for Information Management, Modeling and Simulation (CIMS), Brussels (Belgium)*

Erik Demeulemeester

*KU Leuven, Research Center for Operations Management, Leuven (Belgium)*

Liesje De Boeck

*KU Leuven, Center for Information Management, Modeling and Simulation (CIMS), Brussels (Belgium)*

Jorne Van den Bergh

*KU Leuven, Center for Information Management, Modeling and Simulation (CIMS), Brussels (Belgium)*


---


*Corresponding author

*Email addresses:* philippe.debruecker@kuleuven.be (Philippe De Bruecker), simon.dejaeger@kuleuven.be (Simon De Jaeger), jeroen.belien@kuleuven.be (Jeroen Beliën), erik.demeulemeester@kuleuven.be (Erik Demeulemeester), liesje.deboeck@kuleuven.be (Liesje De Boeck), jorne.vandenbergh@kuleuven.be (Jorne Van den Bergh)

**Abstract**

This paper presents and compares two different solution approaches to optimize the glass collection process at a single intermunicipal authority in Belgium. First, a model enhancement approach is proposed to solve a shift scheduling optimization problem integrating the construction of glass collection routes. Second, a simulation based rolling horizon procedure is designed to determine the possible benefits of providing the glass containers with fill level sensors. The performance of both solution approaches is evaluated and compared using real-life data.

*Keywords:* Model enhancement, waste collection, shift scheduling

## 1. Introduction

The continuous effort within the European Union to maximize the potential for re-use and recycling has drastically increased the complexity of the waste and material collection process in many member states. In addition, the growing tendency towards waste separation at the source (i.e. the separation of waste into different flows at household or firm level) stresses the need for an effective and efficient organization of the collection process. In Belgium, the collection of household packaging waste is an important public service that is outsourced to the private company Fost Plus. Fost Plus promotes, coordinates and finances the selective collection, sorting and recycling of household packaging waste in Belgium. Our focus in this paper is on the glass collection process that is coordinated by Fost Plus. The glass collection itself is not performed by Fost Plus, but is decentralized to different intermunicipal authorities (IAs). These IAs collect the glass and charge Fost Plus with the collection costs which can amount to more than 500,000 Euro per year in one IA. Fost Plus realizes a great share of its profit by selling the recycled waste to other companies. This profit, of course, heavily depends on the costs charged by the IAs for the collection process. Therefore, optimizing the collection processes in the IAs is very important in order to maximize the recycling profits realized by Fost Plus.

A fixed weekly schedule is currently used in each of the IAs that defines the work schedule (comprising the work days and the days off) for each truck driver as well as the collection routes for each driver on each day. In order to optimize the current collection process, Fost Plus wants to consider two different possibilities. First, it wants to consider a second shift type (N shifts) on top of the single current shift type (P shifts) which contains the peak traffic hours on the road. The current P shifts (containing the peak traffic hours) are cheaper than the N shifts (not containing the peak traffic hours) because the P shifts cover the normal daytime working hours (i.e., 9 AM to 5 PM) while the working hours of the N shifts can lie (partially) outside this interval. Prior to any negotiations concerning the definition of the N shifts, the company is primarily interested in the potential benefits of these N shifts. While the P shifts are cheaper than the N shifts, the driving times during the P shifts are on average higher than those during the N shifts. This difference creates a trade-off between higher costs and faster driving times which possibly results in a better workforce schedule with lower weekly labor costs.

Second, Fost Plus wants to analyze the possible benefits of installing and using sensors in all glass containers in order to obtain a real-time view of the fill level of each container. Using this information, collection routes can be constructed on a daily basis in order to avoid wasting driving time to containers with sufficient capacity left (which is unavoidable in the case of a fixed weekly schedule). We refer to the solution obtained with sensor information as a flexible schedule compared to the fixed weekly schedule obtained by the shift scheduling model.

In order to analyze both possibilities, two solution approaches to optimize the glass collection process are proposed and compared. We illustrate the performance of our solution techniques using several test cases based on real-life data. Finally, the proposed models are applied on the real-life data of a single IA.

## 2. Literature review

Constructing a workforce schedule for glass collection is a complicated task. As in all industries, workforce schedules should be constructed according to certain labor union constraints. Each schedule is for example constrained by a maximum time limit and should include a break of a certain duration. However, designing a workforce schedule for glass collection also requires a routing decision. This means that we have to decide on the composition of the collection routes (the routing decision) and the personnel and shift schedules (the scheduling decision) at the same time. Only by combining the routing and the scheduling decision can an optimal result be obtained. However, the integration of the routing decision and the scheduling decision is not straightforward and makes these types of problems very challenging.

The routing decision is an important complicating factor when solving the glass collection problem. The routing problem considered here belongs to the broad class of Vehicle Routing Problems (VRP) (Mes, 2012). In the VRP, a given set of vehicles must deliver goods to a given set of customers such that the overall transportation costs are minimized. Hence, the goal is to construct a set of optimal routes in order to reduce the total traveled distance and possibly the required number of vehicles. In our case, the glass containers represent the customers and the goods to be delivered is empty space. Hence, collecting glass can be seen as filling the containers with air. For an overview of the solution techniques designed for the VRP we refer to Carić and Gold (2008). Recall that we propose two solution approaches to optimize the glass collection process at Fost Plus: a fixed weekly scheduling model combining N and P shifts and a flexible scheduling model based on fill level sensor information. While both approaches contain the VRP, there is an important difference which categorizes them into two different VRP subclasses.

In the first approach, the goal is to find a fixed weekly schedule that is repeated each week during the considered time horizon (e.g., one year). This problem is related to the periodic vehicle routing problem (PVRP) where customers require service on multiple days during a given planning horizon, while the VRP is only concerned with one period. Two decisions have to be made using an integrated or a two-stage approach. First, the weekly service frequency (i.e., how often each customer is served) and service pattern (i.e., which customers will be served on which day of the week) must be determined. Second, a VRP is solved for each day based on the selected customers for that day according to the VRP rules. Francis et al. (2008) present a literature review on the PVRP and its extensions, showing that most of the PVRP research assumes a predetermined service frequency. Only a few researchers incorporate the service frequency as a decision variable in the optimization model. While we also assume a predetermined service frequency (i.e., each container should be visited at least once a week), the service pattern decision is integrated with the routing decision. Our model has three special features which increases the complexity compared to the standard PVRP. First, we incorporate a shift scheduling decision, which is barely investigated in the existing literature in combination with the PVRP. In their literature survey, Ghiani et al. (2014) state that a large percentage of total waste management cost related to waste collection is due to the equipment and the workforce (about 75% according to Shamshiry et al. (2011)). Optimizing the workforce scheduling process can therefore result in significant savings. However, these aspects have not been given much attention in the literature (Ghiani et al., 2014). This is confirmed by Ernst et al. (2004) in their literature review on staff scheduling and rostering where the authors point out the lack of contributions related to personnel and vehicle shift scheduling in the waste management literature. In the literature review of Francis et al. (2008) on the (P)VRP, shift scheduling is not even mentioned. The papers that do consider a combination of staff scheduling and the vehicle routing problem propose a multi-staged approach instead of an integrated approach, or fail to incorporate important routing constraints such as truck capacity constraints (Baudach et al., 2009; Ghiani et al., 2013; Hansmann and Zimmermann, 2009; List et al., 2006). Second, while the standard PVRP only links the different days in the planning horizon in the objective function, seeking the overall minimal transportation costs over all days, we also link the days in the constraints. Like Coene et al. (2010), we consider the scenario where the load of a vehicle at the end of a day needs to be equal to the load of that vehicle at the start of the following day. Furthermore, Francis et al. (2008) state that in the PVRP literature it is assumed that a fraction $\frac{1}{f_i}$ of the total demand has to be delivered to customer $i$ on each visit, with $f_i$ the number of visits required for customer $i$ during the planning horizon. Hence, at each visit, a demand of $w_i = \frac{W_i}{f_i}$ is delivered, with $W_i$ the total demand of customer $i$. This can be a good approach when each visit also involves a delivery and when the daily demand can be assumed to be constant. However, in our problem, we assume that each visit does not necessarily involve emptying the container. Furthermore, we show that strong seasonality effects exist in our data which implies that we cannot assume a constant daily fill rate of glass over the considered time horizon. Hence, our procedure will link all days in the considered time horizon by taking into account the effect of the visits and/or emptying on each day on all succeeding days in the considered time horizon. Third, our problem features intermediate facilities where vehicles can unload (or reload) and thus renew capacity during a route. The PVRP with intermediate facilities is described by Angelelli and Speranza (2002), Kim et al. (2006), Alonso et al. (2008) and Coene et al. (2010). In order to deal with the increased complexity in the PVRP caused by these three elements, we propose a model enhancement heuristic that iteratively combines simulation and optimization.

In the second approach, the goal is to construct a flexible schedule based on fill sensor information. The difference with the previous approach is that no service frequency or service pattern must be determined in this case. Each day, the most urgent containers are first selected based on sensor information. Second, a VRP is solved for the set of selected glass containers. This second approach is not related to the PVRP, but to the Inventory Routing Problem (IRP) which integrates inventory management and vehicle routing (Andersson et al., 2010; Campbell et al., 1998; Coelho et al., 2014). In the IRP, a given set of vehicles must replenish the inventory of a given set of shops over a given planning horizon such that the overall transportation cost is minimized and the total cost of stockouts is minimized. The flexible glass collection scheduling problem can be seen as an IRP since a stockout can be seen as a full container. Since decisions have to be made for each day in the considered time horizon (e.g., one year), two solution approaches are possible. On the one hand, one can consider all days in the planning horizon at once and construct a model that seeks a global optimum since it considers the long term effects of each routing decision. To solve this model, Campbell et al. (1998) propose an integer program. However, in order to keep the integer program computationally tractable, many simplifications and assumptions are required (Campbell et al., 1998), eliminating its property to find the global (over the entire time horizon) optimum. On the other hand, a rolling horizon approach can be much more efficient and effective to solve the IRP (Jaillet et al., 1997). The advantage of a rolling horizon approach is that we do not have to consider the entire time horizon. Instead, it is sufficient to consider only the next few days. Such a rolling horizon approach is adopted by Mes (2012) and Johansson (2006) to solve the dynamic waste collection problem. In the rolling horizon approach for dynamic waste collection, Must Gos (MUGOs) and May Gos (MAGOs) are defined to determine the most useful containers to visit and empty each day. In this paper, we follow a similar approach as Mes (2012) and Johansson (2006) to construct a flexible schedule that is based on fill level sensor information.

## 3. Problem definition

### *3.1. Glass collection procedure*

This section describes the glass collection procedure as it is currently performed by the different IAs. In each IA, a certain number of trucks is available each day for glass collection. Furthermore, each IA employs a certain number of truck drivers that cannot work in the weekend. The truck drivers are paid on a daily basis instead of on an hourly basis. This means that on each week day, each truck driver is paid for 8 hours or has a day off. When a driver has a day off, he is also not paid for that day. Each glass container in the IA is located on a site. Each site can hold one or more glass containers. There are two different types of glass containers: duo containers and standard containers. Duo containers are divided in two equal compartments; one compartment for white glass and one compartment for colored glass. Standard glass containers have only one compartment and can be used for either white glass or colored glass. We make a difference between these two types of containers because of the differences in capacity and the time required to empty the container and collect the glass. Duo containers are overground containers that are smaller compared to the standard containers. Therefore, it takes less time to empty duo containers compared to standard containers that are placed underground. To minimize the glass placed next to the containers when the container is full (i.e., to minimize overflow), Fost Plus demands that each glass container is visited at least once a week.

For each truck, the collection route starts at the depot of the IA. Each truck carries one large container with two compartments, one for colored glass and one for white glass. Leaving the depot, the truck will visit the glass containers as specified by the route in the workforce schedule. While all containers specified in the route must be visited, they will only be emptied when one of the compartments is filled for at least a certain percentage. This percentage is referred to as the collection threshold and is further discussed in Section 5. The truck continues to check and empty glass containers until one of its compartments is getting full. At this time, the truck has to make a trip to the drop-off site of the IA. At the drop-off site, the full container is unloaded from the truck and is replaced with an empty container. Of course, this unloading and loading takes a certain amount of time. When all containers in the route are visited, the truck returns to the depot location. Because a truck is only emptied when one of its compartments is getting full, its final fill level at the end of the day equals the fill level of that truck in the beginning of the next day. Allowing a trip to the drop-off site at the end of the day can, of course, reduce the working time. However, this is not allowed in real life.

### *3.2. Fill rate of glass containers*

The fill level of each compartment in each container on each day depends of course on the routes and the collections that take place on each day. But in the first place, the fill level depends on the fill rate. The fill rate is the amount of glass that is added to a certain container on a specific day. Hence, accurate data regarding the fill rate is very important to make a realistic model. The next paragraph therefore describes our procedure to calculate the fill rate based on glass collection data.

In order to determine the fill rate for each compartment of each container on each day based on the collection data made available by Fost Plus, some data conversions were required. Because some important constraints in our model are expressed in volume (e.g., the capacity of glass containers and the capacity of the trucks), the collection data (given in weight) is converted to $dm^3$. To make these conversions, a conversion of 400 $kg/m^3$ for overground containers and a conversion of 600 $kg/m^3$ for underground containers is assumed. Because an underground container is larger in size, the impact of falling glass and the compression inside the container caused by the weight of the glass is greater. Therefore, the glass inside an underground container is more shattered and a larger conversion factor is assumed. Based on the resulting volumes, the daily fill rate is calculated for each container representing the glass volume that is added (in $dm^3$) in each compartment of the container. We assume that an equal daily amount of glass is added to a container between two consecutive collections. Hence, the daily fill rate is assumed to be constant between two consecutive collections. As a result, it is sufficient to divide the collected amount of glass on a given collection day by the number of days since the last collection day to obtain the daily fill rate. If, for example, 7500 $dm^3$ of white glass was collected on day 7 for a specific container, and that container was emptied on day 2 last time, the daily fill rate is 1500 $dm^3$ for day 3 until day 7.

*3.3. Shift scheduling*

As Fost Plus wants to consider a combination of different shift types (i.e., P shifts and N shifts), we propose a general shift scheduling model to optimize the glass collection process. The goal of the proposed general model is to analyze different scenarios regarding shift costs and driving times. This problem can mathematically be represented by the model presented below.

We first list the sets, along with their associated indices:

| | |
|---|---|
| $d \in D$: | days in the week. With $D = \{1, 2, ..., 5\}$ |
| $w \in W$: | available truck drivers = daily available trucks |
| $i \in I$: | set of glass containers |
| $r \in R$: | set of feasible routes |
| $t \in T$: | set of different shift types |

The coefficients and right hand side constants are presented below:

| | |
|---|---|
| $C_{t,d,w}$: | cost of scheduling a shift of type $t$ on day $d$ for truck $w$ |
| $V_{i,r}$: | = 1 if container $i$ is visited in route $r$; = 0 otherwise |
| $\Theta^{Max}$: | maximum daily average working time (in hours) |

The decision variables are:

| | |
|---|---|
| $x_{t,d,w} \in \{0, 1\}$: | = 1 if a shift of type $t$ is scheduled on day $d$ for truck $w$; = 0 otherwise |
| $\lambda_{r,d,w} \in \{0, 1\}$: | = 1 if route $r$ is used on day $d$ for truck $w$; = 0 otherwise |

We define the following auxiliary variables which completely depend on the former two decision variables:

| | |
|---|---|
| $\tau^{route}_{t,r,d,w}$: | average time required to perform route $r$ on day $d$ with truck $w$ with a shift of type $t$ (in hours) |

The optimization model can be formulated as follows:

Model 1: Shift scheduling model

$$\text{Minimize: } \sum_{t \in T} \sum_{d \in D} \sum_{w \in W} C_{t,d,w} x_{t,d,w} \tag{1}$$

Subject to:

$$\sum_{t \in T} x_{t,d,w} \leq 1, \quad \forall d \in D, \forall w \in W \tag{2}$$

$$\sum_{r \in R} \lambda_{r,d,w} - \sum_{t \in T} x_{t,d,w} = 0, \quad \forall d \in D, \forall w \in W \tag{3}$$

$$\text{if} \quad x_{t,d,w} = 1 \quad \text{then: } \sum_{t' \in T \setminus t} x_{t',d+1,w} = 0, \quad \forall t \in T, \forall w \in W, \forall d \in \{1, ..., 4\} \tag{4}$$

$$\sum_{d \in D} \sum_{w \in W} \sum_{r \in R} V_{i,r} \lambda_{r,d,w} \geq 1, \quad \forall i \in I \tag{5}$$

$$\sum_{t \in T} \sum_{r \in R} \lambda_{r,d,w} \tau^{route}_{t,r,d,w} x_{t,d,w} \leq \Theta^{Max}, \quad \forall d \in D, \forall w \in W \tag{6}$$

$$\tau^{route}_{t,r,d,w} = f \begin{pmatrix} \{\lambda_{r',d',w'} : r' \in R, d' \in D, w' \in W\}, \\ \{x_{t',d',w'} : t' \in T, d' \in D, w' \in W\} \end{pmatrix}, \quad \forall t \in T, \forall r \in R, \forall d \in D, \forall w \in W \tag{7}$$

In the objective function (1) the weekly labor costs are minimized. The total number of shifts required (i.e., $\sum_{t \in T} \sum_{d \in D} \sum_{w \in W} x_{t,d,w}$) is also referred to as the total number of truck days.

Expressions (2) to (7) represent the constraints in the shift scheduling problem. Constraint (2) ensures that there will be at most one shift scheduled on each day for each driver. Constraint (3) shows that each shift should be associated with exactly one collection route $r \in R$. The set of feasible routes $R$ contains all collection routes that meet certain conditions. First, each route $r \in R$ is a tour that begins and ends at the depot location of the IA. Second, a feasible collection route consists of at least one glass container and at most 60 glass containers. According to Fost Plus, this maximum number of containers keeps the schedule manageable and allows the truck drivers to become familiar with the collection routes.

According to the labor union requirements, there should be at least a certain amount of time between two consecutive shifts. Because two different shift types (e.g., an N shift and a P shift) cover different working hours, they cannot succeed each other. For example, an N shift cannot be followed by a P shift or vice versa. This is ensured in our model by Constraint (4) which represents the shift succession constraint. Because none of the workers can work in the weekend, the shift succession constraint is only concerned with the week days.

To minimize the amount of glass that is put next to the glass container (i.e., to minimize overflow) because the container is full, Fost Plus determined that each glass container should be visited at least once a week. This is ensured by Constraint (5). Recall that visiting a container does not necessarily mean that this container is emptied. Only when one of its compartments is filled for at least the collection threshold, both compartments are emptied.

Finally, the labor union requirements also state that a worker can work (on average) for at most 8 hours in a shift including a 30 minute break. With the average working time, we mean the average working time for that day and truck over all weeks in the considered time horizon. The considered time horizon can be as small as one week and as large as 1 year (since we have data for one year). This labor union requirement is represented by Constraint (6). Hence, the Right Hand Side (RHS) of this constraint equals 7.5 hours ($\Theta^{Max} = 7.5$). The Left Hand Side (LHS) calculates the total average working time of the respective route for that particular day and truck (driver).

Constraint (7) shows that $\tau^{route}_{t,r,d,w}$ is a function of all routing variables $\lambda_{r',d',w'}$ with $r' \in R$, $d' \in D$, $w' \in W$ and all shift scheduling variables $x_{t',d',w'}$ with $t' \in T$, $d' \in D$ and $w' \in W$. Hence, it is a function of all the routing and scheduling decisions in the model. To make things more clear, we can write $\tau^{route}_{t,r,d,w}$ as the sum of three separate times as follows:

$$\tau^{route}_{t,r,d,w} = \Theta^{driving}_{t,r} + \tau^{collection}_{r,d} + \tau^{drop\text{-}off}_{t,r,d,w} \tag{8}$$

According to Equation (8), the total average working time ($\tau^{route}_{t,r,d,w}$) consists of three parts, namely the driving time ($\Theta^{driving}_{t,r}$), the collection time ($\tau^{collection}_{r,d}$) and the drop-off time ($\tau^{drop\text{-}off}_{t,r,d,w}$). The first part ($\Theta^{driving}_{t,r}$) only represents the driving time from the depot location to the first container in route $r$, the driving time between all consecutive containers in route $r$ and the driving time from the last container in route $r$ back to the depot location. Hence, $\Theta^{driving}_{t,r}$ only depends on the sequence of glass containers in route $r$ and is independent of the fill level of the containers and the fill level of the truck. This means that $\Theta^{driving}_{t,r}$ is independent from the routing decisions in the model. Therefore, $\Theta^{driving}_{t,r}$ is not a decision variable, but a parameter of the chosen route with index $r$. The subscript $t$ shows that, depending on the scheduled shift type $t$ for that day and truck, the location-to-location driving times are different. However, we assume that these driving times will be exactly the same in each week in the considered time horizon. Hence, we assume deterministic driving times.

Note that, based on the former definition of $\Theta_{t,r}^{driving}$, the sum of the three terms in Equation (8) would account for more time than is actually required. Recall that $\Theta_{t,r}^{driving}$ consists of the driving time from the depot to the first container in route $r$, the driving time between all consecutive containers in route $r$ and the driving time from the last container in route $r$ back to the depot location. However, when a drop-off is required, the truck immediately goes from container $K$ to the drop-off location and returns to the next container $K+1$ in the route (with $K$ the value of the index of a container in the route). Hence, the trip from container $K$ to $K+1$ (trip $K \rightarrow K+1$), which is accounted for in $\Theta_{t,r}^{driving}$, gets replaced by trip $K \rightarrow$ drop-off $\rightarrow K+1$. To make sure that we do not account for more driving time than is actually required, these excess driving times (from $K$ to $K+1$) should be subtracted. As $\Theta_{t,r}^{driving}$ is preferred to remain a constant, the excess driving time is accounted for in $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$. This is very important in order for the proposed solution technique to perform well and is further explained in Section 4.4.

The second part of the total average working time is the collection time $\tau_{r,d}^{collection}$. This time does not have a subscript $t$ which means that it is independent of the scheduled shift type. This is because this time does not contain any driving time. The collection time only consists of the *average* collection time for all containers in route $r$. Hence, it is the time required to empty containers when one of its compartments is filled for at least the collection threshold on day $d$. Therefore, the collection time $\tau_{r,d}^{collection}$ depends of course on the route decisions for the other days. This is because a certain glass container can also be collected by this or another truck on another day. Therefore, $\tau_{r,d}^{collection}$ depends on the values of the decision variables in the optimization model and, hence, is not a constant like $\Theta_{t,r}^{driving}$. While the driving time is independent of the day, the collection time does depend on the day during which the route is executed because the fill level of each container can be different on each day. Note also that $\tau_{r,d}^{collection}$ is an average over all weeks in the considered time horizon while we assumed $\Theta_{t,r}^{driving}$ to be constant over all weeks.

The last part of the total average working time is the drop-off time $\tau_{t,r,d,w}^{drop\text{-}off}$. Just as $\tau_{r,d}^{collection}$, $\tau_{t,r,d,w}^{drop\text{-}off}$ is also considered to be a decision variable because the drop-off time depends on the routing decision in the optimization model. Furthermore, just as $\tau_{r,d}^{collection}$, $\tau_{t,r,d,w}^{drop\text{-}off}$ is an average over all weeks in the considered time horizon. First, $\tau_{t,r,d,w}^{drop\text{-}off}$ consists of the time that is required to interrupt the route when the truck is getting full. Based on his experience, the truck driver decides whether the next container in the route can be emptied without causing the truck to overflow. When, according to the driver, it is not possible to empty the next container without causing an overflow of the truck, a trip to the drop-off location is required. Second, the drop-off time also consists of the total average time that is required to unload the truck at the drop-off location and load it again with an empty container. The position of the different drop-offs in the route (which determines the total driving time to and from the drop-off location) and the number of required drop-offs (which determines the total loading and unloading time), not only depend on the fill level of the containers on day $d$, but also on the initial fill level of truck $w$ at the beginning of that day. Recall that the initial fill level of a truck depends on the route during the previous shift for that truck. Just as $\Theta_{t,r}^{driving}$, $\tau_{t,r,d,w}^{drop\text{-}off}$ contains the subscript $t$ because the driving times during the trips to and from the drop-off depend on the scheduled shift type.

The definition of each of the three terms in Equation (8) shows that only $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ are variables in $\tau_{t,r,d,w}^{route}$ that depend on the decisions made by the optimization model, while $\Theta_{t,r}^{driving}$ is a constant belonging to the route with index $r$ and the shift type with index $t$. Therefore, both $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ are a function of the set of all routing variables $\lambda_{r',d',w'}$ and shift scheduling variables $x_{t',d',w'}$ with $t' \in T$, $d' \in D$ and $w' \in W$:

$$
\begin{aligned}
\tau_{r,d}^{collection} &= f'\begin{pmatrix} \{\lambda_{r',d',w'} : r' \in R, d' \in D, w' \in W\}, \\ \{x_{t',d',w'} : t' \in T, d' \in D, w' \in W\} \end{pmatrix}, \quad \forall r \in R, \forall d \in D \\
\tau_{t,r,d,w}^{drop\text{-}off} &= f''\begin{pmatrix} \{\lambda_{r',d',w'} : r' \in R, d' \in D, w' \in W\}, \\ \{x_{t',d',w'} : t' \in T, d' \in D, w' \in W\} \end{pmatrix}, \quad \forall t \in T, \forall r \in R, \\
& \qquad \forall d \in D, \forall w \in W
\end{aligned}
$$

Based on Equation (8), we can now rewrite Constraint (6) as Constraint (9):

$$
\begin{aligned}
&\sum_{t\in T}\sum_{r\in R} \lambda_{r,d,w}\left[\Theta_{t,r}^{driving} + \tau_{t,r,d,w}^{drop-off}\right] x_{t,d,w} + \\
&\sum_{r\in R} \lambda_{r,d,w}\tau_{r,d}^{collection} \leq \Theta^{Max}, \qquad \forall d \in D, \forall w \in W \qquad (9)
\end{aligned}
$$

As our problem contains the vehicle routing problem (VRP), our problem is NP-hard. This makes it very difficult to solve to optimality in a reasonable amount of time for realistic sized instances. But even without the difficulty added by the VRP, so even if we assume a very small set of possible routes $R$, solving model 1 is not straightforward. The difficulty of solving model 1 clearly lies in the definition of $\tau_{t,r,d,w}^{route}$ and in particular of $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$. An explicit definition of $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ implies an explicit formulation of the functions $f'()$ and $f''()$. This would require a complete mathematical description of the entire collection process in each week of the considered time horizon linking the fill level of each container and the route decision for each truck driver on each day with each other. Instead of such a complex explicit mathematical formulation, simulation is much better suited for these cases. We evaluate a certain workforce schedule with a simulation model resulting in a value for $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ without the requirement for an explicit mathematical formulation of $f'()$ and $f''()$ in the optimization model. Since $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ are also averages on day $d$ over all weeks in the considered time horizon, simulation also removes the need to model each single week explicitly in the optimization model.

Section 4 describes the iterative model enhancement procedure combining simulation and optimization to solve model 1.

### *3.4. Sensors and a flexible schedule*

Until now, we only focused on a fixed weekly schedule that determines for each driver the collection route and shift schedule. Hence, every week, the same collection routes are executed and the same containers are visited on the same day. Recall that visiting a container does not necessarily mean that this container is also emptied. A container is only emptied when one of its compartments is filled for at least the collection threshold. When a container is visited while it was not necessary (i.e., when none of its compartments is filled for at least the collection threshold), time is wasted.

We now consider a scenario where we want to avoid visiting containers that are not very critical in order to avoid wasting time. Based on sensor information, we can get information about the fill level of each container, allowing us to avoid such useless trips. However, using sensor information to search for critical containers and constructing routes accordingly, means that we cannot longer assume a fixed weekly schedule. In fact, a completely flexible schedule is required which entails that each route for each truck on each day in each week can be different. Fost Plus only started to experiment very recently with such sensors and is interested in the possible savings that can be obtained.

Based on the fill rate data, a solution procedure is suggested in Section 4.5 that analyzes the possible benefits of sensors and a flexible schedule. The obtained results presented in Section 5 are not only useful to analyze the benefits of sensors, but also to evaluate the performance of the solution procedure to obtain a fixed weekly schedule.

## 4. Methodology

In this section we investigate the two possible ways to improve the current glass collection process as it is currently in use in the IA under study in order to reduce the costs that are charged to Fost Plus. First, we investigate the possibility to extend the current workforce schedule (consisting of only shifts of type P) with shifts of type N. As was already discussed in Section 3.3, the difficulty of the shift scheduling problem as described in model 1 lies in the explicit definition of $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ which requires the complex explicit formulation of $f'()$ and $f''()$. Instead of such an explicit formulation, simulation is much better suited in these cases. In Sections 4.1 to 4.4, we discuss the model enhancement procedure that iteratively combines simulation and optimization to approximate and solve model 1.

Second, we move away from a fixed schedule to a flexible schedule and investigate the possibility to exploit the use of sensors. This is the subject of Section 4.5.

### *4.1. Model Enhancement*

In this paper, we use a technique called "model enhancement" to solve the shift scheduling problem. The term *model enhancement* (ME) was used by Bachelet and Yon (2007) to indicate a different way of combining simulation and optimization. It can be seen as a decomposition method like Benders' decomposition as it enhances an optimization model based on simulation results by adding constraints (Benders, 1962). While most optimization-simulation couplings focus on improving the objective function evaluated from simulation (like the simulation optimization approach), ME still focuses on optimizing the theoretical objective function. It tries to improve the solution provided by a mathematical model by the use of simulation (Bachelet and Yon, 2007). In their paper, Bachelet and Yon (2007) assume that in practice several modeling simplifications are needed to construct the mathematical optimization model to solve a real-life problem. In most cases, the resulting model therefore fails to give a correct representation of reality. In the ME framework, simulation is used to enhance the mathematical model and to improve the realism and applicability of the solution.

One way to solve model 1 presented above is to estimate $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ for each day, truck, route and shift type and to use this estimate in the optimization model. This way, variables $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ are transformed into parameters $\Theta_{r,d}^{collection}$ and $\Theta_{t,r,d,w}^{drop\text{-}off}$, removing the need for Constraint (7) in model 1. Hence, $\Theta_{r,d}^{collection}$ and $\Theta_{t,r,d,w}^{drop\text{-}off}$ are independent of the shift and route decisions in the optimization model. Constraint (9) can therefore be rewritten as:

$$\sum_{t\in T}\sum_{r\in R}\lambda_{r,d,w}\left[\Theta_{t,r}^{driving}+\Theta_{t,r,d,w}^{drop\text{-}off}\right]x_{t,d,w}+\sum_{r\in R}\lambda_{r,d,w}\Theta_{r,d}^{collection}\leq\Theta^{Max},\quad\forall d\in D,\ \forall w\in W$$

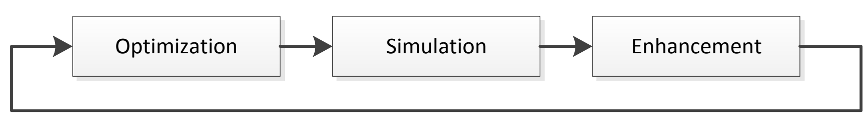
This is of course a simplification that allows us to approximate $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$

without explicitly modeling them in the optimization model. However, we do not know whether we will obtain a good or even feasible solution this way. Because of its simplicity, this approach is called straightforward optimization by Bachelet and Yon (2007).

A better way to approximate $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ is to use ME which allows to iteratively improve the initial estimate of $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ with the help of simulation. In a ME model, optimization and simulation are used in an iterative procedure to enhance the optimization model. Hence, the goal is to exploit the benefits of simulation which allows us to solve the problem without an explicit formulation of $f'()$ and $f''()$. Just as with straightforward optimization, ME allows us to replace the variables $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ by parameters. However, in contrast to straightforward optimization, the estimate of the parameter values is enhanced during each ME iteration based on a simulation run.

The three phases of the ME procedure are shown in Figure 1 and are discussed below.

Figure 1: Model enhancement algorithm




*4.2. Phase I: Optimization*

The first phase in the ME procedure is the optimization phase. In order to apply ME to solve model 1, variables $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ are transformed into parameters. Therefore, $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ must be made independent of the decisions made by the optimization model (i.e., the shift scheduling and the routing decision). First, this means that the route index $r$ in the subscripts of $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ should be removed (since the model decides on the route). Since there can only be one route for each day and truck (see Constraint (2)), the route index $r$ can be replaced by the truck index $w$. Second, making $\tau_{r,d}^{collection}$ and $\tau_{t,r,d,w}^{drop\text{-}off}$ independent of the decisions made by the optimization model means that also the shift type index $t$ can be removed (since the model decides on the scheduled shift types).

Hence, both variables are transformed into parameters as follows:

$$\tau_{r,d}^{collection} \quad \rightarrow \quad \Theta_{d,w}^{collection}$$
$$\tau_{t,r,d,w}^{drop\text{-}off} \quad \rightarrow \quad \Theta_{d,w}^{drop\text{-}off}$$

Because both $\Theta_{d,w}^{collection}$ and $\Theta_{d,w}^{drop\text{-}off}$ will now be parameters in the optimization model with the same indices $d$ and $w$, we can add them together as $\Theta_{d,w}^{c\&d}$:

$$\Theta_{d,w}^{c\&d} = \Theta_{d,w}^{collection} + \Theta_{d,w}^{drop\text{-}off}, \qquad \forall d \in D, \forall w \in W$$

We can now formulate model 2, the adjusted shift scheduling model, as an approximation of model 1. As shown below, model 2 features the same objective function as model 1. Model 2 only differs from model 1 regarding Constraint (6) (which is the same as Constraint (9)) and Constraint (7). These two constraints are replaced by Constraint (10) as follows:

Model 2: Adjusted shift scheduling model

$$\text{Minimize: } \sum_{t\in T}\sum_{d\in D}\sum_{w\in W} x_{t,d,w}C_{t,d,w}$$

Subject to:

Constraints (2) to (5)

$$\sum_{t\in T}\sum_{r\in R}\lambda_{r,d,w}\Theta_{t,r}^{driving}x_{t,d,w} + \sum_{r\in R}\lambda_{r,d,w}\Theta_{d,w}^{c\&d} \leq \Theta^{Max}, \qquad \forall d \in D, \forall w \in W \quad (10)$$

*4.2.1. Solution technique*

In order to solve the problem formulated in model 2, tabu search is used. Tabu Search (TS) is a local search combinatorial optimization technique developed by Glover and Laguna (1997). It starts from an initial solution and tries to improve that solution by making moves through a specified search space to the neighborhood of the current solution. The move that improves the solution the most is then made. This procedure continues until a stopping criterion is met. Each time a move is made, that move becomes *tabu* for a certain number of iterations. The set of moves that are made tabu are collected in a tabu list. By holding these moves in a tabu list for a certain number of iterations, the algorithm avoids making loops through the search space and tries to escape from local optima.

During the first enhancement iteration (i.e., the first loop through the three phases in Figure 1), the simulation phase (phase II) is only executed after the optimization phase for the first time. Therefore, there were no previous simulation results to define $\Theta_{d,w}^{c\&d}$. $\Theta_{d,w}^{c\&d}$ is therefore initialized to an initial estimate for each day $d \in D$ and each truck $w \in W$. During the next enhancement iterations, $\Theta_{d,w}^{c\&d}$ is defined and enhanced during the enhancement phase (phase III).

In order to optimize the solution to the problem described in model 2, a tabu search algorithm is designed that swaps the shift type of each scheduled shift. Tabu search is ideal for this because swapping one P shift for an N shift will not immediately decrease the objective value in most cases. In fact, the objective value will most likely increase with a single swap. In most cases, several swaps are required before the objective value will decrease. By making recent swaps tabu, it is possible to escape from this local optimum and to search for a better shift schedule. In each optimization phase, the tabu search algorithm is applied to the shift schedule obtained during the previous optimization phase.

The idea behind swapping P by N shifts is to reduce the number of truck days (which is the total number of required shifts). Depending on the cost of an N shift, this swap move can also decrease the objective value (the total weekly labor costs). The idea behind swapping N shifts by P shifts is to decrease the total weekly costs while maintaining the same number of truck days.

In order to evaluate each swap move, a collection route is constructed for each scheduled shift in the adjusted shift schedule until all containers are included in a collection route. The constructed collection routes are first assigned to the schedule of the first truck and only then to the second truck. Hence, we make a distinction between the two trucks as we first try to fill the schedule of the first truck and only use the second truck when the first truck appears to be insufficient to visit all the containers. This distinction between the two trucks during the optimization phase removes symmetry and avoids the efficiency problems that symmetry would cause. When the shift schedule consists of too few scheduled shifts in order to construct collection routes such that all containers are included, an extra P shift is added to the end of the shift schedule and the procedure continues. Recall that during each optimization phase the tabu search algorithm is applied to the shift schedule obtained during the previous optimization phase. Since there is no previous shift schedule during the first enhancement iteration, a certain number of P shifts is added until all containers can be included in a collection route. The collection routes are constructed using a cheapest insertion heuristic in order to reduce the required number of shifts while minimizing the driving times in each route. This way we are compressing the collection routes in view of reducing the required number of shifts. Hence, when it appears that fewer shifts are required in order to include all containers in a collection route than the number of shifts that is currently scheduled, the unnecessary shifts are removed. Note that compressing the collection routes in view of reducing the required number of shifts means that the total driving time of the obtained solution does not necessarily decrease during the enhancement iterations. Therefore, the best shift schedule with the lowest total driving time is saved during the enhancement procedure. Minimizing the driving times can be of interest because of its direct link with fuel costs.

When the collection routes are constructed, the succession constraints are checked and the shift schedule is adjusted if necessary. When the succession constraints are violated because of the swap move, the necessary days off are inserted (if possible) to render the shift schedule feasible again. This step also allows to remove unnecessary days off from the shift schedule resulting from the swap move.

After building and optimizing the collection routes and adding or removing shifts from the adjusted shift schedule, the cost of the shift schedule is calculated according to the objective function (1) in order to numerically evaluate each move. Note that it is only during the construction of the collection routes that Constraint (10) is checked to ensure a feasible shift schedule.

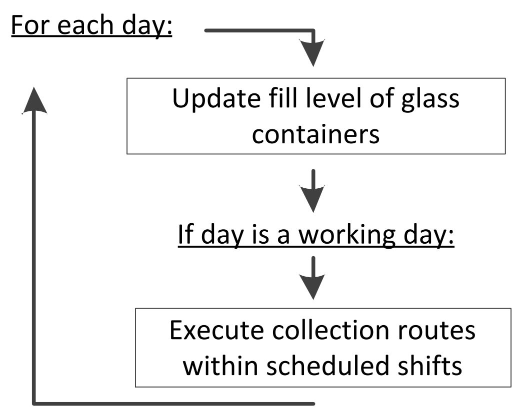
During each optimization phase, 100 tabu search iterations are performed. The optimization phase will always result in a feasible result with respect to Constraint (10). While the objective value of the obtained result is not necessarily lower than the one during the previous enhancement iteration, the assumptions made in the model (i.e., the approximation of the real average collection and drop-off times by $\Theta_{d,w}^{c\&d}$) are getting more realistic. Eventually, this will result in good feasible solutions with respect to Constraints (6) and (7) in model 1.

### *4.3. Phase II: Simulation*

The second phase in the enhancement procedure is the simulation phase. It is during this phase that a simulation model is ran in order to evaluate the solution obtained in phase I. Figure 2 shows a schematic overview of the steps in the simulation process.

As Figure 2 shows, the fill level of all glass containers is only updated at the beginning of each day based on the fill rate (see Section 3.2). This means that the simulation model has a precision of one day. Hence, we assume that the containers are filled at the beginning of each day with the total amount of glass that was dropped in the container during the previous day. During the rest of the day, no glass is added to the container. When the current day in the simulation is a working day (i.e., not a weekend day), the scheduled collection routes are executed within the scheduled shifts. The execution of the collection routes is simulated by letting a truck with a certain capacity follow the route within the scheduled shift. Recall that the initial fill level of the truck depends on the final fill level during the previous shift of that truck. The truck visits each container according to the collection route and collects the glass of the visited container if one of its compartments

Figure 2: Schematic overview of the simulation process




is filled for at least the collection threshold. When the truck is full (which depends on the judgment of the driver as is explained in Section 3.3), the collection route is interrupted by a trip to the drop-off location.

The goal of the simulation is to evaluate the solution obtained during the optimization phase with respect to the total average working time on each day $d$ for each truck $w$. Hence, the simulation gives us the real (under the assumptions of the simulation) value of the LHS of Constraint (6) in model 1. Since Constraint (10) in model 2 is only an approximation of Constraint (6) in model 1, Constraint (6) may or may not be satisfied according to the simulation results. To make sure that Constraint (10) is a good approximation of Constraint (6), causing both Constraint (10) and Constraint (6) to be satisfied, the simulation information is used to enhance the approximation. This is done during the enhancement phase.

### *4.4. Phase III: Enhancement*

During the enhancement phase, the estimate of $\Theta^{c\&d}_{d,w}$ in model 2 is enhanced based on the simulation results during phase II. The following notations are used to formulate the enhancement function:

$\Theta_{d,w}^{route^{Simulation}}$: average time required according to the simulation to perform the route scheduled on day $d$ with truck $w$ with the scheduled shift type on day $d$ for truck $w$ (in hours). Recall that we define the total route time as the sum of the driving, collection and drop-off time.

$\Theta_{d,w}^{(c\&d)^{Simulation}}$: average time required according to the simulation to empty glass containers and to perform the required drop-offs in the route scheduled on day $d$ with truck $w$ with the scheduled shift type on day $d$ for truck $w$ (in hours).

Next, we define $\delta$ as the index of the enhancement iterations, with $\delta = 1$ the first enhancement iteration. Using $\delta$, all notations in model 2 can be indexed for each enhancement iteration. This way, we can define:

$\Theta_{d,w}^{route^{\delta}}$: total estimated average time (resulting from the optimization of model 2) required to perform the route scheduled on day $d$ with truck $w$ with the scheduled shift type on day $d$ for truck $w$ during enhancement iteration $\delta$ (in hours). Recall that we define the total route time as the sum of the driving, collection and drop-off time. Hence, this equals the value of the LHS of Constraint (10) in the optimal solution for model 2 during enhancement iteration $\delta$.

$\Theta_{d,w}^{(c\&d)^{\delta}}$: estimate during enhancement iteration $\delta$ of the average time required to empty glass containers and to perform the required drop-offs in the route scheduled on day $d$ with truck $w$ with the scheduled shift type on day $d$ for truck $w$ (in hours).

Based on the former definitions, we can now state the enhancement function to enhance the previous value of $\Theta_{d,w}^{c\&d}$ as follows:

$$\text{If} \quad \left(\Theta_{d,w}^{route^{Simulation}} \leq \Theta^{Max}\right):$$

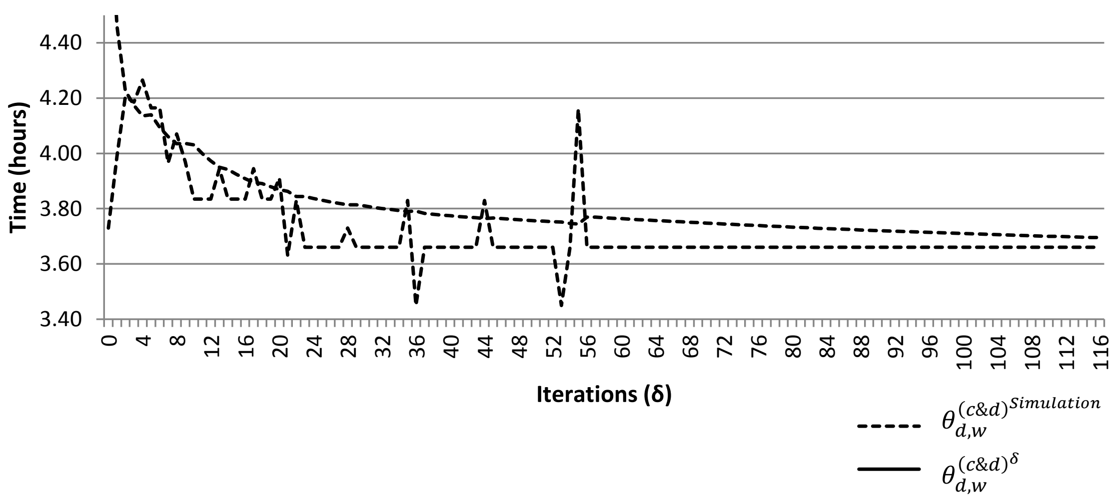
$$\Theta_{d,w}^{(c\&d)^{\delta+1}} = \frac{\Theta_{d,w}^{(c\&d)^{\delta}} \cdot \delta}{\delta + 1} + \frac{\Theta_{d,w}^{(c\&d)^{Simulation}}}{\delta + 1}$$

Else:

$$\Theta_{d,w}^{(c\&d)^{\delta+1}} = \Theta_{d,w}^{(c\&d)^{\delta}} + \Theta^{Max} - \Theta_{d,w}^{route^{\delta}} + 0.01 \tag{11}$$

In enhancement function (11), $\Theta_{d,w}^{(c\&d)^{\delta+1}}$ is the estimate during the next enhancement iteration of the average time required to empty glass containers and to perform the required drop-offs in the route scheduled on day $d$ with truck $w$ with the scheduled shift type on day $d$ for truck $w$. If $\Theta_{d,w}^{route^{Simulation}} \leq \Theta^{Max}$, a moving average is calculated causing the impact of the simulation to diminish over time. This way, we seek after the convergence of $\Theta_{d,w}^{c\&d}$. As $\Theta_{d,w}^{c\&d}$ converges and stabilizes, $\Theta_{d,w}^{(c\&d)^{Simulation}}$ can also converge and stabilize. In most cases, $\Theta_{d,w}^{c\&d}$ will therefore converge to $\Theta_{d,w}^{(c\&d)^{Simulation}}$ just as in Figure 3.

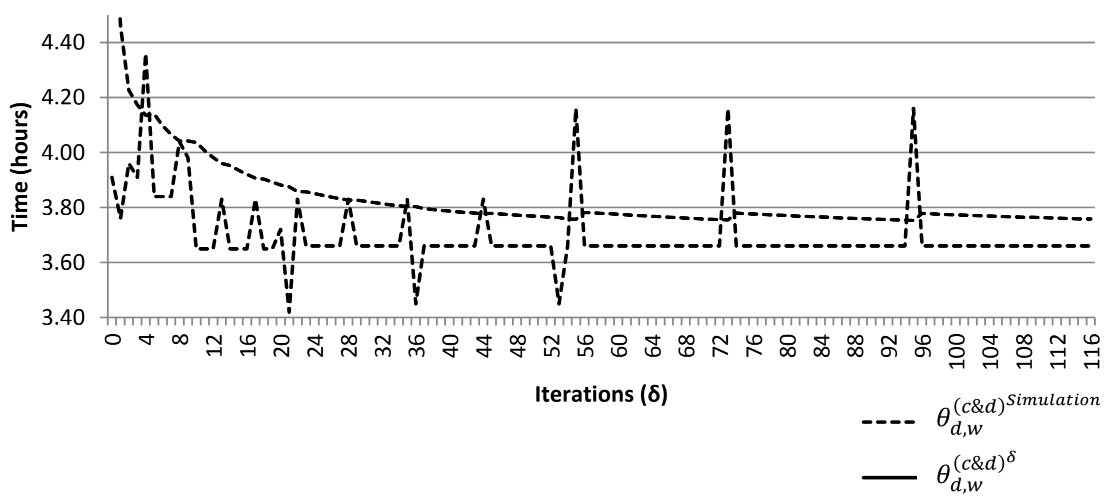
Figure 3: Example of the convergence of $\Theta_{d,w}^{c\&d}$ to $\Theta_{d,w}^{(c\&d)^{Simulation}}$




However, it is possible that $\Theta_{d,w}^{c\&d}$ will not converge to $\Theta_{d,w}^{(c\&d)^{Simulation}}$ as is shown in Figure 4. Figure 4 shows an example of the progress of $\Theta_{d,w}^{c\&d}$ and $\Theta_{d,w}^{(c\&d)^{Simulation}}$ over the course of 116 enhancement iterations for a certain day and truck. As the graph in Figure 4 shows, $\Theta_{d,w}^{c\&d}$ descends towards $\Theta_{d,w}^{(c\&d)^{Simulation}}$ during the first 54 iterations. However, during iteration 55, $\Theta_{d,w}^{(c\&d)^{Simulation}}$ suddenly peaks over $\Theta_{d,w}^{c\&d}$. It is at this point that $\Theta_{d,w}^{c\&d}$ is small enough to add one extra glass container in the route of this truck according to model 2. Recall, however, that model 2 does not immediately take into account the possible extra collection and drop-off time that is required to add this extra glass container. In order for model 2 to take this extra time into account, we have to wait for the feedback from the simulation evaluation. As the peaks during iterations 55, 73 and 95 show, it appears to be impossible to add an extra container to the route when the extra collection and drop-off time is accounted for. In other words, the total average working time is greater than $\Theta^{Max}$ during these peaks. Hence, $\Theta_{d,w}^{c\&d}$ can never be as low as $\Theta_{d,w}^{(c\&d)^{Simulation}}$ in this case since we do not immediately take into account the extra required collection and drop-off time.

When $\Theta_{d,w}^{route^{Simulation}} > \Theta^{Max}$, we want to escape as quickly as possible from this infeasible situation. Therefore, the second part (the Else case) of the enhancement function (11) is applied. In order to avoid spending too much time in the infeasible situation waiting for $\Theta_{d,w}^{(c\&d)^{\delta}}$ to increase with the help of the moving average, $\Theta_{d,w}^{(c\&d)^{\delta}}$ is immediately increased by the spare time during iteration $\delta$ ($\Theta^{Max} - \Theta_{d,w}^{route^{\delta}}$). To ensure that the extra container cannot remain in the route, an extra 0.01 hours is added. This makes sure that the container is removed from the route during iteration $\delta + 1$. This explains why $\Theta_{d,w}^{(c\&d)^{Simulation}}$ peaks at iterations 55, 73 and 95, and drops immediately at the next iteration. This procedure renders the solution feasible more quickly and assures more stability during the next iterations which allows to find better solutions more quickly.

Figure 4: Example of the convergence of $\Theta_{d,w}^{c\&d}$




To determine the value of $\Theta_{d,w}^{(c\&d)^{Simulation}}$ during the enhancement procedure, we do not directly use the collection and drop-off time supplied by the simulation. In Section 3.3 we argued that the drop-off time, as it is defined in Section 3.3, should be reduced by some driving time in order for Equation (8) to be exact. Recall that the driving time ($\Theta_{t,r}^{driving}$) consists of the driving time from the depot location to the first container in route $r$, the driving time between all consecutive containers in route $r$ and the driving time from the last container in route $r$ back to the depot location. However, when a drop-off is required, the truck does not immediately go to the next container, but goes first to the drop-off location. Hence, $\Theta_{t,r}^{driving}$ contains some unnecessary container-to-container driving times when one or more drop-offs were required. To ensure that we only account for the necessary driving time, $\Theta_{d,w}^{(c\&d)^{Simulation}}$ should be reduced by these excess driving times. Therefore, we use the equation presented below to determine the value of $\Theta_{d,w}^{(c\&d)^{Simulation}}$ based on $\Theta_{d,w}^{route^{Simulation}}$ and the driving times resulting from the optimization phase. $\Theta_{d,w}^{route^{Simulation}}$ contains the driving times for a trip to and from the drop-off ($K \rightarrow$ *drop-off* $\rightarrow K+1$), but not the driving time for the unnecessary trip $K \rightarrow K+1$, while $\sum_{t \in T} \sum_{r \in R} \lambda_{r,d,w} \Theta_{t,r}^{driving} x_{t,d,w}$ does contain the driving time for the unnecessary trip $K \rightarrow K+1$. Hence, enhancing $\Theta_{d,w}^{(c\&d)^{\delta}}$ based on $\Theta_{d,w}^{(c\&d)^{Simulation}}$ and using it in Constraint (10) ensures that we only account for the necessary driving time:

$$\Theta_{d,w}^{(c\&d)^{Simulation}} = \Theta_{d,w}^{route^{Simulation}} - \sum_{t \in T} \sum_{r \in R} \lambda_{r,d,w} \Theta_{t,r}^{driving} x_{t,d,w}.$$

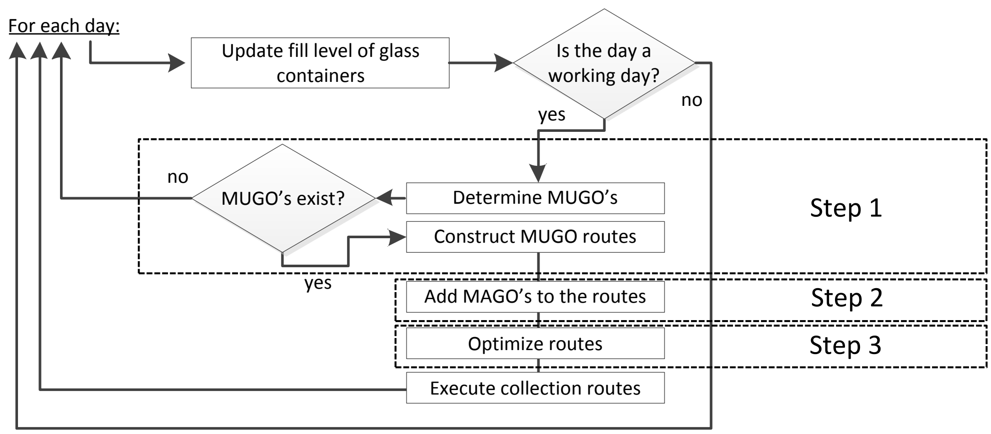
### *4.5. Sensors and flexible routes*

Until now, we only focused on a fixed weekly schedule that determines for each driver the collection route and the shift schedule. Hence, every week, the same collection routes are executed and the same containers are visited on the same day. Using a fixed schedule will often result in visiting a container while it was not necessary (i.e., when none of its compartments is filled for at least the collection threshold). In order to avoid wasting time, we can construct a flexible schedule where we avoid such useless trips using sensor information about the fill level of each container on each day.

A flexible schedule is based on sensor information about the fill level of each container on each day. Using sensor information, we assume that we know the exact amount of glass that each compartment of each container holds. Using historical data, we also assume that we know the amount of glass that will be added during the following days. This way, we can determine the set of containers that should be emptied today in order to avoid that a container will overflow.

Containers that should be emptied today are referred to as Must Gos (MUGOs). All other containers are referred to as May Gos (MAGOs). While MUGOs *must* be emptied today in order to avoid overflow, MAGOs may or may not be emptied. From Monday to Wednesday, we define each container that will overflow during the next two days as a MUGO. Hence, we assume a buffer of one day. On Friday, each container that will overflow during the next three days is defined as a MUGO because no collections can take place during the weekend. Because Friday would become very busy in this case, a buffer of two days is assumed to define MUGOs on Thursday. Recall that each glass container is located on a site and that a site can hold one or more glass containers. Hence, when a certain container is getting full, people can continue to drop glass in the other containers located on the same site. In such cases, these containers are not assumed to overflow and are therefore not regarded as MUGOs.

Figure 5: Schematic overview of the rolling horizon procedure




In order to determine the required number of truck days in order to empty all critical containers, a rolling horizon approach is used that is similar to the simulation approach outlined in Section 4.3. However, instead of using fixed routes, we will now build collection routes on a daily basis based on sensor information. The rolling horizon procedure is presented in Figure 5. Just as in Section 4.3, the fill level of all glass containers is only updated at the beginning of each day. During the rest of the day, no glass is added to the container. When the current day in the simulation is a working day (i.e., not a weekend day), we loop through three steps to build the collection routes. Finally, the collection routes are executed. The execution of the collection routes is simulated by letting a truck with a certain capacity follow the route. Recall that the initial fill level of the truck depends on the final fill level during the previous shift of that truck. The difference with the collection procedure from Section 4.3 is that each visited container will always be emptied in this case. When the truck is full (which depends on the judgment of the driver as is explained in Section 3.3), the collection route is interrupted by a trip to the drop-off location.

To build the collection routes, we start by determining the set of MUGOs at the beginning of each work day. When there are no MUGOs that day, no collection routes will be constructed. When there are some MUGOs, we first construct one or more collection routes such that all MUGOs are included in a route. Second, the most urgent MAGOs are added to these routes. Finally, the resulting routes are optimized to reduce the driving time. These three steps are discussed below.

<u>Step 1: Constructing MUGO routes</u>

When constructing the routes in a flexible schedule scenario, we have to ensure that at least all MUGO containers can be emptied every day. The first step is therefore to construct collection routes only for the MUGO containers. This is done based on the nearest neighbor heuristic. It is during this first step that the minimal number of trucks (taking into account the number of available trucks) is determined that is required on this day to empty all MUGOs. Hence, a truck will only be used when there is at least one MUGO container to empty. When one truck can empty all MUGO containers on a given day, no extra truck is required.

Step 2: Adding MAGOs to the collection routes

Based on the sensor information, we know exactly how long each route will take (taking into account the required drop-offs). Therefore, we can now add MAGOs to the routes as long as we do not exceed $\Theta^{Max}$ working hours. MAGOs are added to the route of a certain truck at a certain position using the cheapest insertion heuristic. MAGOs are inserted in the routes according to a sorted list (sorted by the number of days until the container will overflow) such that the most urgent MAGO is added first. Hence, we define the urgency of a container based on the number of days until one of the compartments of the container will overflow. Again, we take into account the required drop-off time during each insertion and keep filling up the routes until no more MAGOs can be added without exceeding $\Theta^{Max}$ working hours. Using this procedure, each of the visited containers will be quite full. Therefore, we do not use the same collection threshold as in the fixed schedule but assume that each visited container will also be emptied.

Note that during the second step, we insert MAGOs based on a sorted list where we give priority to the most urgent MAGOs without taking into account the location of the MAGO container. Only when two MAGOs are equally urgent, is the additional driving distance taken into account. Adding the most urgent MAGO to the route can therefore drastically increase the total driving time of the collection route. However, since the most urgent MAGO of today can become the MUGO of tomorrow, we increase the chance to reduce the number of MUGOs to zero during the next day, potentially decreasing the required number of truck days. As this rolling horizon procedure only minimizes the number of MUGOs and, consequently, the required number of truck days on the next day, it lacks a long-term optimization vision. A different MAGO selection decision could, for example, lead to fewer MUGOs and fewer truck days in the long run. Without resorting to a long-term optimization model in combination with the rolling horizon procedure, different MAGO selection rules can be considered. According to Mes (2012), two different possibilities are usually used in an IRP setting to determine which customers should be visited first. The most common way is to use the ratio of the fill level to the additional travel time required to visit this container (Francis et al., 2008; Golden et al., 1984). Another frequently used technique is to look at the ratio of urgency to the additional travel time (Campbell et al., 1998). The last three columns in Table 1 show a comparison of our MAGO selection technique (urgency) with the latter two techniques based on the resulting number of truck days. The three techniques are investigated under six different test settings representing the entire planning period (one year), a particular busy period and a more calm period in two different intermunicipal authorities (IA1 and IA2). The results show that our selection technique prevails as the resulting number of truck days (and, hence, the resulting labor costs) found with the other techniques is never lower than for our technique.

Table 1: Comparison of three MAGO selection techniques based on the resulting number of truck days

| test setting | urgency | $\frac{\text{fill level}}{\text{extra distance}}$ | $\frac{\text{urgency}}{\text{extra distance}}$ |
|---|---|---|---|
| IA1: 1 year period | 6.40 | 6.51 | 6.42 |
| IA2: 1 year period | 5.75 | 6.06 | 5.93 |
| IA1: busy period | 7.75 | 7.75 | 8.00 |
| IA2: busy period | 6.00 | 6.50 | 6.50 |
| IA1: calm period | 6.19 | 6.27 | 6.19 |
| IA2: calm period | 5.69 | 5.97 | 5.88 |
| Average | 6.30 | 6.51 | 6.49 |

Note: The test settings are constructed using data from two different intermunicipal authorities (IA1 and IA2).

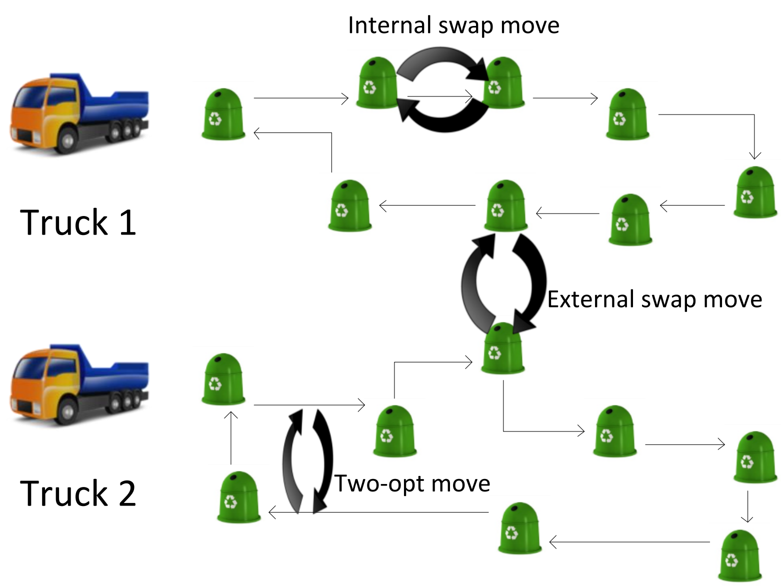
Step 3: Optimizing the collection routes

After the collection routes are constructed, the routes are optimized in order to reduce the total driving distance (directly related to the fuel costs) without the intention of adding more MAGOs. Instead, the time made available by the optimization is used as a buffer against uncertainty. In order to optimize the route, three types of swap moves are consecutively applied within a fastest descent framework.

1. *Internal swap moves:* Containers are swapped within the route of the same truck.
2. *External swap moves:* Containers are swapped between the routes of different trucks.
3. *Two-opt moves:* Arcs are swapped within the route of the same truck.

Figure 6 shows an example of each of the three moves. The internal swap move swaps two containers in the route of the same truck. The external swap move swaps two containers between the routes of two trucks. The two-opt move swaps two arcs within the route of the same truck. Two-opt moves are a classic heuristic optimization technique for routing problems. First, the two respective arcs are removed from the route such that the route is divided into two parts. Next, the direction of all the arcs is reversed in one of these parts. Finally, the two parts are joined again with two different arcs.

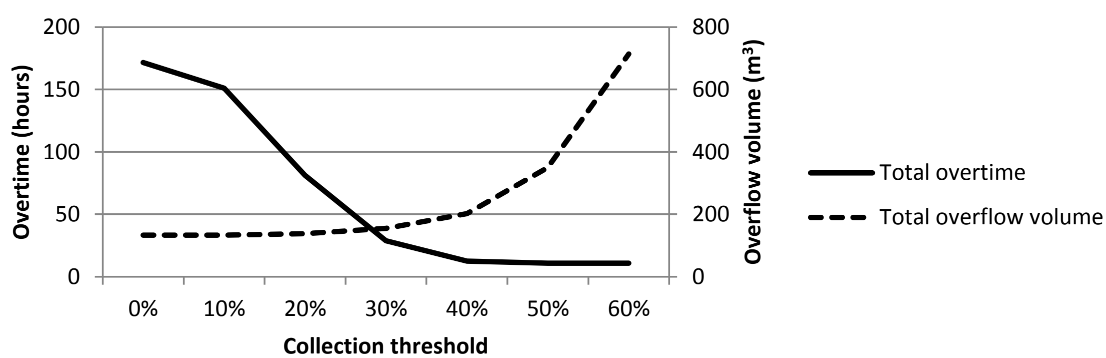
Figure 6: Example of the three moves during the route optimization




## 5. Results and discussion

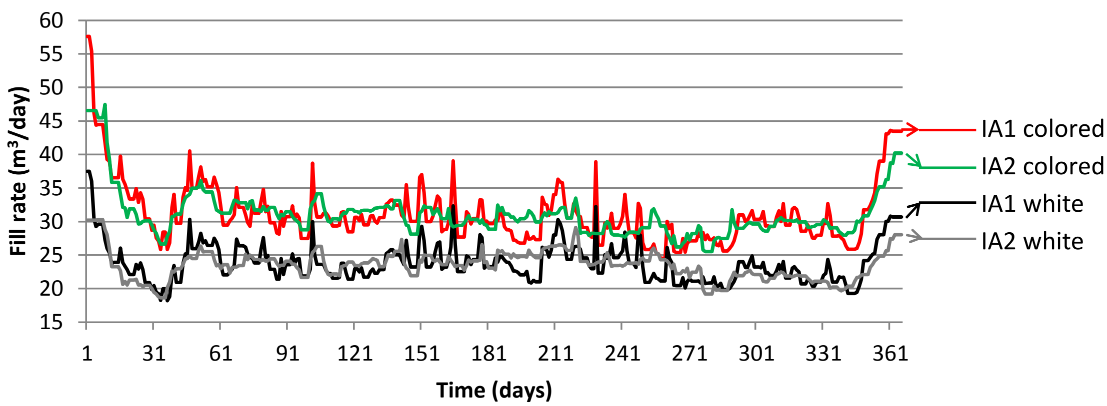
In order to analyze the possible savings for the glass collection process at Fost Plus, we focus on one specific IA that charges Fost Plus with the collection costs. The IA under focus is selected based on the availability of data, the reliability of the available data and the geographic location of the IA (a fairly urbanized and busy area). Because of confidentiality reasons, we cannot give the name of the IA under study, but we can present some of its key properties. In the IA under study, two trucks are available each day for glass collection. Each truck can hold 13 $m^3$ of white glass and 17 $m^3$ of colored glass. Furthermore, two truck drivers are employed that cannot work in the weekend. In total, there are more than 300 containers in the IA, located on more than 200 different sites. Most of the containers are duo containers, while only a small fraction consists of standard containers. We assume that duo containers take 6 minutes to empty and can hold 1.675 $m^3$ glass in each of its compartments, while standard containers take 12 minutes and can hold 4 $m^3$ glass. At the drop-off site, loading and unloading is assumed to take 20 minutes. Finally, we assume a collection threshold of 40%, meaning that visited containers are only emptied when one of its compartments is filled for at least 40%. This percentage is based on a trade-off between overtime and glass overflow volume. Using the right collection threshold percentage is critical in the fixed scheduling problem. As the same schedule is repeated every week, each container must be scheduled at least once in the weekly schedule. Therefore, each container will be visited at least once a week. However, in most cases, it is not necessary to empty each container every week. When each visited container would also be emptied (i.e., a collection threshold of 0%), the glass overflow will be minimal. But as the resulting working time will increase because of the increased number of emptyings and drop-offs, the overtime will also increase. Therefore, it is better to wait until the containers are filled with more glass before emptying them. This can be observed in Figure 7 which shows the trade-off between the total overtime and the glass overflow volume for different collection thresholds. At a fill level of more than 40%, the glass overflow volume will increase very fast, while the total overtime almost reaches its minimum at 40%. Hence, 40% seems to be a good trade-off between overtime and glass overflow volume. The results in Figure 7 are obtained by simulating a fixed weekly schedule with only P shifts during one year. While this figure shows the results for only one schedule and one IA, the same trend can also be observed for other schedules and even other IAs. This analysis, however, goes beyond the scope of this paper.

Figure 7: Collection threshold analysis




Based on real-life collection data of one year, the daily fill rate of each compartment of each container is calculated according to the procedure described in Section 3.2. Figure 8 shows a graphical representation of the aggregated daily fill rate (in $m^3$) for white and colored glass for two IAs, where IA1 is the IA under study. The fill rate shows a similar trend for both IAs and shows large peaks during the New Year period. The ratio between white and colored glass is also very similar in both IAs.

Figure 8: Aggregated daily fill rate during one year




*5.1. Theoretical results*

Before we apply the ME model to the real-life setting, we first analyze the performance of the model regarding feasibility and solution quality. To get a good view on the performance of the proposed solution technique, we create a set of test cases that are based on the first 4 weeks of the year. Recall that this considered time frame falls within the peak period for glass collection. Therefore, we can assume that the fill rate during the rest of the year will not be higher than during these four weeks. Hence, finding a good feasible schedule for this period is more challenging which makes the considered time frame very interesting.

The proposed ME procedure (developed in C++) allows to solve model 1 in order to analyze the possible advantages of introducing a second shift type. Recall that two different shift types are considered: a P shift containing the peak traffic hours with a standard shift cost and an N shift not containing the peak traffic hours with a higher cost. While the N shift is more expensive than the P shift, the average driving speed is higher during the N shift compared to the P shift. We further assume that there are two trucks and two truck drivers available to work each day. To allow for a thorough analysis, five different scenarios are constructed regarding the average driving speed during an N shift compared to a P shift. For each of the different speed-up scenarios, six different scenarios are constructed regarding the cost premium for an N shift compared to a P shift. The results of the ME algorithm for these 30 different test cases are presented in Table 2 using the following definitions:

$\sigma$: factor ($\leq 1$) that is used to multiply the driving times in a P shift to obtain the driving times in an N shift.

$C_N$: cost of an N shift. The cost of a P shift ($C_P$) is always assumed to be 1.

$\theta_w^{estimate}$ $= Avg_{d \in D} \left\{ \Theta_{d,w}^{route} \right\}$ = estimate in model 2 of the average working hours for truck $w$ over all active truck days for truck $w$.

$\theta_w^{simulation}$ $= Avg_{d \in D} \left\{ \Theta_{d,w}^{route^{Simulation}} \right\}$ = average working hours for truck $w$ over all active truck days for truck $w$ according to the simulation.

We test the model for five different $\sigma$ values ranging from 0.9 to 0.5 and six different $C_N$ values ranging from 1.1 to 1.6. Columns 2 and 3 of Table 2 show the value of $\sigma$ and $C_N$ respectively. The index of each of the 30 test cases is shown in Column 1. The results presented in columns 4 to 9 are obtained by running the ME algorithm for 100 iterations. This takes on average 10 minutes for each test case. Columns 4 to 6 show the required number of truck days, the required number of P and N shifts and the resulting weekly costs of the shift schedule. Columns 8 to 9 show for each of the 2 trucks (indexed in column 7) the average estimated working hours and the average simulated working hours. At the bottom of Table 2, we show the average and the standard deviation of columns 8 and 9.

### *5.1.1. Feasibility*

The first criterion for a good solution is of course the feasibility of the solution with respect to all the constraints in model 1 (i.e., Constraints (2) to (7)). In this section we only focus on the feasibility with respect to Constraints (6) and (7) as these are the constraints that are approximated by constraint 10 in model 2 with the aid of ME. All other constraints in model 1 are also present in model 2 and will therefore be satisfied by definition. Hence, we want the real (i.e., simulated over the four weeks in our considered time frame) total average working hours of the obtained solutions to be less than or equal to 7.5 hours ($\Theta^{Max}$) on each day for each truck. The convergence of $\Theta_{d,w}^{c\&d}$ (possibly, but not necessarily to $\Theta_{d,w}^{(c\&d)^{Simulation}}$) which is aimed at during the enhancement phase in the ME procedure, should ensure that Constraints (6) and (7) in model 1 are satisfied. When $\Theta_{d,w}^{c\&d}$ converges to $\Theta_{d,w}^{(c\&d)^{Simulation}}$, Constraint 10 is a good approximation of Constraints (6) and (7). Hence, when Constraint 10 is satisfied, Constraints (6) and (7) are also satisfied. As we have shown in Section 4.4, $\Theta_{d,w}^{c\&d}$ will not always converge to $\Theta_{d,w}^{(c\&d)^{Simulation}}$. If it does not converge, $\Theta_{d,w}^{c\&d}$ will be greater than $\Theta_{d,w}^{(c\&d)^{Simulation}}$. Therefore, Constraints (6) and (7) are satisfied when Constraint (10) is satisfied.

While the tabu search algorithm during the optimization phase in the ME procedure ensures that Constraint (10) in model 2 is never violated, it is the enhancement phase that has to make sure that Constraints (6) and (7) are satisfied. We found that our ME procedure is able to find a feasible result for each test case (satisfying Constraints (6) and (7)). Column 9 of Table 2 then also shows that the real (simulated) average total working time (over all active truck days) $\theta_w^{simulation}$ is less than or equal to 7.5 hours for each test case. As can be observed in these two tables, $\theta_w^{estimate}$ and $\theta_w^{simulation}$ always lie fairly close to each other. Furthermore, there is no statistically significant difference between these two averages (two sided p value of 0.2994).

#### *5.1.2. Solution quality*

The second criterion of a good solution is the solution quality in terms of the obtained objective value (the weekly costs). The obtained objective value for each of the test cases can be found in column 6 of Table 2. In this section we first analyze the quality of the obtained solutions based on the properties of an optimal solution (see Sections 5.1.2.1 and 5.1.2.2). Second, we analyze the quality based on a comparison with the results of a flexible schedule (see Section 5.1.2.3).

Table 2: Theoretical results Fost Plus

| Case | $\sigma$ | $C_N$ | Truck days | Shifts | Weekly costs | Truck $w$ | $\theta_w^{estimate}$ | $\theta_w^{simulation}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.9 | 1.1 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 2 | 0.9 | 1.2 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 3 | 0.9 | 1.3 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 4 | 0.9 | 1.4 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 5 | 0.9 | 1.5 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 6 | 0.9 | 1.6 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 7 | 0.8 | 1.1 | 7 | 0P + 7N | 7.70 | 1 | 7.38 | 7.33 |
| | | | | | | 2 | 6.54 | 6.39 |
| 8 | 0.8 | 1.2 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 9 | 0.8 | 1.3 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 10 | 0.8 | 1.4 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 11 | 0.8 | 1.5 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 12 | 0.8 | 1.6 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 13 | 0.7 | 1.1 | 7 | 5P + 2N | 7.20 | 1 | 7.39 | 7.39 |
| | | | | | | 2 | 7.40 | 7.38 |
| 14 | 0.7 | 1.2 | 7 | 5P + 2N | 7.40 | 1 | 7.39 | 7.39 |
| | | | | | | 2 | 7.40 | 7.38 |
| 15 | 0.7 | 1.3 | 7 | 5P + 2N | 7.60 | 1 | 7.39 | 7.39 |
| | | | | | | 2 | 7.40 | 7.38 |

Theoretical results Fost Plus Cont.

| Case | $\sigma$ | $C_N$ | Truck days | Shifts | Weekly costs | Truck $w$ | $\theta_w^{estimate}$ | $\theta_w^{simulation}$ |
|---|---|---|---|---|---|---|---|---|
| 16 | 0.7 | 1.4 | 7 | 5P + 2N | 7.80 | 1 | 7.39 | 7.39 |
| | | | | | | 2 | 7.40 | 7.38 |
| 17 | 0.7 | 1.5 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 18 | 0.7 | 1.6 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 19 | 0.6 | 1.1 | 6 | 1P + 5N | 6.50 | 1 | 7.34 | 7.33 |
| | | | | | | 2 | 7.20 | 7.21 |
| 20 | 0.6 | 1.2 | 6 | 1P + 5N | 7.00 | 1 | 7.34 | 7.33 |
| | | | | | | 2 | 7.20 | 7.21 |
| 21 | 0.6 | 1.3 | 6 | 1P + 5N | 7.50 | 1 | 7.34 | 7.33 |
| | | | | | | 2 | 7.20 | 7.21 |
| 22 | 0.6 | 1.4 | 7 | 5P + 2N | 7.80 | 1 | 7.33 | 7.33 |
| | | | | | | 2 | 7.33 | 7.12 |
| 23 | 0.6 | 1.5 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 24 | 0.6 | 1.6 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| 25 | 0.5 | 1.1 | 6 | 2P + 4N | 6.40 | 1 | 7.36 | 7.36 |
| | | | | | | 2 | 6.98 | 6.98 |
| 26 | 0.5 | 1.2 | 6 | 2P + 4N | 6.80 | 1 | 7.36 | 7.36 |
| | | | | | | 2 | 6.98 | 6.98 |
| 27 | 0.5 | 1.3 | 6 | 2P + 4N | 7.20 | 1 | 7.36 | 7.36 |
| | | | | | | 2 | 6.98 | 6.98 |
| 28 | 0.5 | 1.4 | 6 | 2P + 4N | 7.60 | 1 | 7.36 | 7.36 |
| | | | | | | 2 | 6.98 | 6.98 |
| 29 | 0.5 | 1.5 | 6 | 2P + 4N | 8.00 | 1 | 7.36 | 7.36 |
| | | | | | | 2 | 6.98 | 6.98 |
| 30 | 0.5 | 1.6 | 8 | 8P + 0N | 8.00 | 1 | 7.36 | 7.26 |
| | | | | | | 2 | 6.94 | 6.94 |
| | | | | | | Average: | 7.20 | 7.16 |
| | | | | | | St.dev.: | 0.22 | 0.20 |

*5.1.2.1. Properties of an optimal solution*

In Table 2, we see that only P shifts are used to construct the shift schedule when $C_N$ is very high (i.e., 1.6 in our case). In such cases, the benefit of using an N shift (i.e., faster driving times) does not outweigh the higher costs. Define $\Lambda_t^{opt.}$ as the minimal number of truck days required to obtain a feasible solution with only shifts of type $t$. Hence, $\Lambda_P^{opt.}$ is the minimal number of truck days required to find a feasible solution with only P shifts. Starting from $\Lambda_P^{opt.}$, we can now ask ourselves if it is possible to reduce the total weekly costs by reducing the number of truck days by using a certain number of N shifts. To analyze this possibility, we first determine the value of $\Lambda_N^{opt.}$. When $\Lambda_N^{opt.} < \Lambda_P^{opt.}$, we propose that it is possible to reduce the weekly costs by combining N and P shifts, depending on the value of $C_N$. Hence, starting from $\Lambda_P^{opt.}$, we can also ask ourselves if is possible to reduce the weekly costs by using a certain number of N shifts for each value of $C_N$. We propose that when $C_N$ decreases, we can find lower weekly costs by decreasing the number of truck days by using more N shifts. Defining $\Delta_{C_N}^{opt.}$ as the minimal number of truck days required to find a feasible solution with $C_N$ defined as the cost of an N shift, the previous statements can be translated in Proposition 1 and Proposition 2. With $C_N^1$ and $C_N^2$ defined as two values for $C_N$, we state Proposition 1 as follows:

Proposition 1:

$\forall\ C_N^1, C_N^2 \mid (C_P < C_N^1 < C_N^2)$: $\Delta_{C_N^1}^{opt.} \leq \Delta_{C_N^2}^{opt.}$.

Proposition 1 states that the optimal number of truck days increases or remains the same when the cost of an N shift increases. Hence, the optimal number of truck days cannot decrease when the cost of an N shift increases. To prove Proposition 1 (and Proposition 2), we first state and prove Lemma 1 and Lemma 2 based on the previous definitions as follows:

Lemma 1:

$\forall\ \Delta_{C_N}^{opt.} \mid (C_N > C_P)$: $\exists!\ \rho^{opt.} \mid (\rho^{opt.} = \sum_{d\in D}\sum_{w\in W} X_{N,d,w}^{opt.})$.

Proof Lemma 1:

Lemma 1 shows that for each optimal number of truck days $\Delta_{C_N}^{opt.}$ (with $C_N > C_P$), there exists exactly one minimal number of N shifts ($\rho^{opt.}$) that is required to obtain $\Delta_{C_N}^{opt.}$ truck days. We prove Lemma 1 by contradiction. When there would exist a second, larger $\rho^{opt.}$ together with the same $\Delta_{C_N}^{opt.}$, it means that the second solution contains more N shifts, resulting in higher costs. The second solution can, therefore, not be an optimal solution.

When there would exist a second, smaller $\rho^{opt.}$ together with the same $\Delta_{C_N}^{opt.}$, it means that the first solution contains more N shifts, resulting in higher costs. The first solution can, therefore, not be an optimal solution. Hence, only one minimal number of N shifts exists for each optimal number of truck days $\Delta_{C_N}^{opt.}$. □

This does, however, not mean that there can only be one optimal number of truck days for each given value of $C_N$. Based on Lemma 1 and using the same kind of proof, we can also state that for each minimal number of N shifts, there exists exactly one optimal number of truck days. Since there exists exactly one $\rho^{opt.}$ for each $\Delta_{C_N}^{opt.}$, we can rewrite $\rho^{opt.}$ as $\rho^{opt.}_{\Delta^{opt.}_{C_N}}$. With $\rho^{opt.}_{\Delta^{opt.}_{C_N}}$ defined as the minimal number of N shifts that is required to obtain $\Delta_{C_N}^{opt.}$ truck days, we can state Lemma 2 as follows:

<u>Lemma 2:</u>

$$\begin{cases} \forall\ \Delta^{opt.}_{C^1_N}, \Delta^{opt.}_{C^2_N} \mid (\Delta^{opt.}_{C^1_N} = \Delta^{opt.}_{C^2_N})\&(C^1_N > C_P)\text{: } \rho^{opt.}_{\Delta^{opt.}_{C^1_N}} = \rho^{opt.}_{\Delta^{opt.}_{C^2_N}} \\ \forall\ \Delta^{opt.}_{C^1_N}, \Delta^{opt.}_{C^2_N} \mid (\Delta^{opt.}_{C^1_N} < \Delta^{opt.}_{C^2_N})\&(C^1_N > C_P)\text{: } \rho^{opt.}_{\Delta^{opt.}_{C^1_N}} > \rho^{opt.}_{\Delta^{opt.}_{C^2_N}} \end{cases}$$

Hence: $\forall\ \Delta^{opt.}_{C^1_N}, \Delta^{opt.}_{C^2_N} \mid (\Delta^{opt.}_{C^1_N} \leq \Delta^{opt.}_{C^2_N})\&(C^1_N > C_P)\text{: } \rho^{opt.}_{\Delta^{opt.}_{C^1_N}} \geq \rho^{opt.}_{\Delta^{opt.}_{C^2_N}}$.

<u>Proof Lemma 2:</u>

Lemma 2 first states that the minimal number of required N shifts remains unchanged when the optimal number of truck days also stays the same. This is a direct consequence of Lemma 1. Hence, there can only be one optimal number of required N shifts for each optimal number of truck days. Second, Lemma 2 states that the minimal number of required N shifts decreases when the optimal number of truck days increases. This can be proven by contradiction. Consider a scenario with two solutions where the second solution contains more truck days and more N shifts compared to the first solution. We prove that the second solution can never be optimal in this case. Note that we do not have to incorporate the case where the number of N shifts is equal in both solutions because this is impossible for optimal solutions when the optimal number of truck days is different (see Lemma 1). When the number of P shifts in the second solution is also greater than or equal to the number of P shifts in the first solution, it is easy to see that the second solution can never be optimal. When the first solution with fewer or the same number of P shifts and fewer N shifts is feasible, it is also feasible for other values of $C_N$. Furthermore, this first solution will always result in lower costs compared to the second solution containing more N shifts and more or the same number of P shifts.

When the second solution with more truck days contains more N shifts, but fewer P shifts, we can give a similar proof to show that the second solution can never be optimal in this case. Note that each P shift in a feasible shift schedule can be replaced by an N shift without rendering the schedule infeasible. An N shift can, of course, not always be replaced by a P shift because the average driving speed in a P shift is lower than in an N shift. In our scenario, we can therefore replace the P shifts in the first solution with N shifts until the number of P shifts in the first solution matches the number of P shifts in the second solution. Since the second solution contains more truck days, the number of N shifts in the adjusted first solution (containing the same number of P shifts as in the second solution) will always be smaller than in the second solution. As the adjusted first solution is still feasible for all values of $C_N$, it will always result in a smaller cost than the second solution which contains the same number of P shifts, but has more N shifts. Hence, the second solution can never be optimal.

Third, Lemma 2 states that from the first two statements it follows that when the optimal number of truck days increases or stays the same, the minimal number of required N shifts decreases or stays the same. □

Note that Lemma 2 is only concerned with the *optimal* number of truck days for a certain value of $C_N$ and not with all possible numbers of truck days. Hence, not all possible integer numbers of truck days between $\Lambda_N^{opt.}$ and $\Lambda_P^{opt.}$ are necessarily part of the set of optimal numbers of truck days.

Proposition 2:

$$\forall\, C_N^1, C_N^2 \mid (C_P < C_N^1 < C_N^2)\text{: } \rho^{opt.}_{\Delta^{opt.}_{C_N^1}} \geq \rho^{opt.}_{\Delta^{opt.}_{C_N^2}}.$$

Conditional proof Proposition 2:
Proposition 2 states that the minimal number of N shifts required in order to obtain a given optimal number of truck days will decrease (or stay constant) when the cost of an N shift increases. Hence, under the assumption that Proposition 1 is valid, Proposition 2 is also valid because of Lemma 2.

Proof Propositions 1 and 2:
In order to prove Propositions 1 and 2, we first give a formal definition of the function describing the optimal weekly costs. We can write the optimal weekly costs in function of the cost of an N shift as in Expression (12). Note that we now use the function variable $c$ instead of $C_N$ to model the cost of an N shift. This is explained in the next paragraph.

$$Weekly\ costs^{opt.}(c) = \rho^{opt.}_{\Delta_c^{opt.}} c + (\Delta_c^{opt.} - \rho^{opt.}_{\Delta_c^{opt.}})C_P \qquad (12)$$

From the definition of the optimal weekly cost function in Expression (12), we see that the function will be linear in $c$ as long as $\Delta_c^{opt.}$ stays constant. The results in Table 2 show that the number of truck days makes discrete increments when the cost of an N shift increases (for a constant value of $\sigma$). Only for the case where $\sigma = 0.9$, the number of truck days stays constant when the cost of an N shift changes. This illustrates that the optimal weekly cost function is composed of different linear partial cost functions with breakpoints for each change in the optimal number of truck days. This means that the optimal weekly cost function is a piecewise linear function formed by connecting the intersections of at most $\Lambda_P^{opt.} - \Lambda_N^{opt.}$ consecutive (sorted according to the respective number of truck days) partial cost functions and confined by the horizontal line $\Lambda_P^{opt.} \cdot C_P$. Hence, for each optimal number of truck days, a unique partial cost function exists, describing the weekly costs in function of $c$. As the optimal number of truck days depends on the cost of an N shift, while the optimal number of truck days is fixed for each partial cost function, we cannot use $\Delta_c^{opt.}$ in the definition of the partial cost functions. Therefore, we make a difference between $C_N$ (constant) and $c$ (function variable) to fix the optimal number of truck days as $\Delta_{C_N}^{opt.}$. This way, the partial cost functions can be defined as in Expression (13).

$$Weekly\ costs^{opt.^{\Delta_{C_N}^{opt.}}}(c) = \rho_{\Delta_{C_N}^{opt.}}^{opt.} c + (\Delta_{C_N}^{opt.} - \rho_{\Delta_{C_N}^{opt.}}^{opt.}) C_P \tag{13}$$

Without making assumptions with respect to the value of the cost of an N shift, we can now describe the general properties of the optimal weekly cost function (see Expression (12)) in order to prove Propositions 1 and 2. For the range of possible $c$ values (with $c > C_P$), a set of (unique) optimal truck days exists such that the elements of this set can be ordered by increasing value. Defining these ordered elements as $\Delta_{C_N^1}^{opt.}$, $\Delta_{C_N^2}^{opt.}$, $\Delta_{C_N^3}^{opt.}$, ..., we can make the following general statement based on Lemma 2.

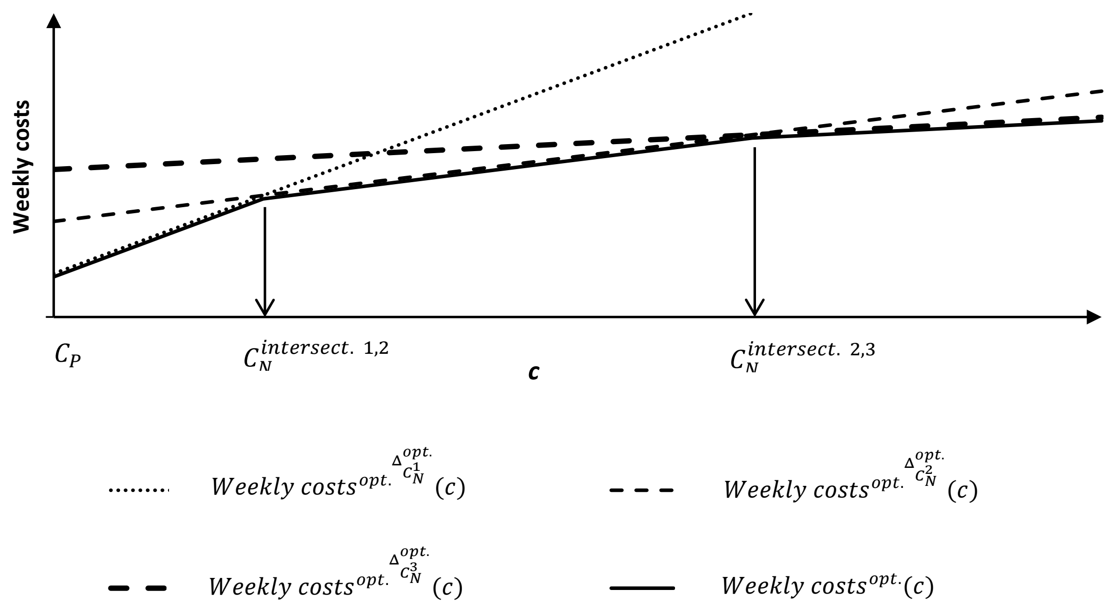
If $\Delta_{C_N^1}^{opt.} < \Delta_{C_N^2}^{opt.} < \Delta_{C_N^3}^{opt.} < \ldots$ , then $\rho_{\Delta_{C_N^1}^{opt.}}^{opt.} > \rho_{\Delta_{C_N^2}^{opt.}}^{opt.} > \rho_{\Delta_{C_N^3}^{opt.}}^{opt.} > \ldots$ .

As each optimal number of truck days corresponds with a partial cost function, the respective partial cost functions can be defined as follows:

$$\begin{cases} Weekly\ costs^{opt.^{\Delta_{C_N^1}^{opt.}}}(c) = \rho_{\Delta_{C_N^1}^{opt.}}^{opt.} c + (\Delta_{C_N^1}^{opt.} - \rho_{\Delta_{C_N^1}^{opt.}}^{opt.}) C_P \\ Weekly\ costs^{opt.^{\Delta_{C_N^2}^{opt.}}}(c) = \rho_{\Delta_{C_N^2}^{opt.}}^{opt.} c + (\Delta_{C_N^2}^{opt.} - \rho_{\Delta_{C_N^2}^{opt.}}^{opt.}) C_P. \\ Weekly\ costs^{opt.^{\Delta_{C_N^3}^{opt.}}}(c) = \rho_{\Delta_{C_N^3}^{opt.}}^{opt.} c + (\Delta_{C_N^3}^{opt.} - \rho_{\Delta_{C_N^3}^{opt.}}^{opt.}) C_P. \\ \qquad \ldots \end{cases}$$

Figure 9 shows the relationship between the optimal weekly cost function (represented by a solid line) and the partial cost functions (represented by the dashed lines). As the optimal weekly cost function resembles the part of a wall that encloses the end of a pitched roof, it is further referred to as the *gable* function. None of the partial cost functions lie below this gable function. The partial cost functions are further referred to as the *beam* functions as they resemble the beams of the roof. As can be observed in Figure 9, the gable function features a breakpoint at each intersection of two consecutive (sorted according to the respective number of truck days) beam functions, indicating a change in the optimal number of truck days. The remainder of this section seeks to describe the properties of the gable and beam functions in order to prove the generality of Figure 9 and, consequently, prove Propositions 1 and 2.

Figure 9: General representation of three partial cost functions (beam functions)




We first investigate the intercept and slope of the beam functions. In general, the following inequalities are valid for the beam functions:

$$\forall\ Weekly\ costs^{opt.\ \Delta^{opt.}_{C_N^1}}(c),\ Weekly\ costs^{opt.\ \Delta^{opt.}_{C_N^2}}(c),\ Weekly\ costs^{opt.\ \Delta^{opt.}_{C_N^3}}(c),$$

$$...\quad |\ (\Delta^{opt.}_{C_N^1} < \Delta^{opt.}_{C_N^2} < \Delta^{opt.}_{C_N^3} < ...)\&(\rho^{opt.}_{\Delta^{opt.}_{C_N^1}} > \rho^{opt.}_{\Delta^{opt.}_{C_N^2}} > \rho^{opt.}_{\Delta^{opt.}_{C_N^3}} > ...):$$

$$\Delta^{opt.}_{C_N^1} - \rho^{opt.}_{\Delta^{opt.}_{C_N^1}} < \Delta^{opt.}_{C_N^2} - \rho^{opt.}_{\Delta^{opt.}_{C_N^2}} < \Delta^{opt.}_{C_N^3} - \rho^{opt.}_{\Delta^{opt.}_{C_N^3}} < ...\ .$$

As $\Delta^{opt.} - \rho^{opt.}$ corresponds with the intercept of a beam function, the following statements are valid:

$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^1_N}}(c) <$$
$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^2_N}}(c) <$$
$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^3_N}}(c) < \ldots\ .$$

Moreover, the following is also valid:

$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^1_N}}(c) \text{ with } c = C_P <$$
$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^2_N}}(c) \text{ with } c = C_P <$$
$$\text{intercept of } Weekly\ costs^{opt. \Delta^{opt.}_{C^3_N}}(c) \text{ with } c = C_P < \ldots\ .$$

The latter statement is true because the intercept with $c = C_P$ equals $\Delta^{opt.}_{C^1_N}$, $\Delta^{opt.}_{C^2_N}$, $\Delta^{opt.}_{C^3_N}$, ... respectively and $\Delta^{opt.}_{C^1_N} < \Delta^{opt.}_{C^2_N} < \Delta^{opt.}_{C^3_N} < \ldots$ . Therefore, none of the beam functions will intersect before $c = C_P$. As $\rho^{opt.}_{\Delta^{opt.}_{C^1_N}}$, $\rho^{opt.}_{\Delta^{opt.}_{C^2_N}}$, $\rho^{opt.}_{\Delta^{opt.}_{C^3_N}}$, ... corresponds with the slope of the respective beam function, we see that:

$$\text{slope of } Weekly\ costs^{opt. \Delta^{opt.}_{C^1_N}}(c) >$$
$$\text{slope of } Weekly\ costs^{opt. \Delta^{opt.}_{C^2_N}}(c) >$$
$$\text{slope of } Weekly\ costs^{opt. \Delta^{opt.}_{C^3_N}}(c) > \ldots\ .$$

Based on the former properties, we can now prove Propositions 1 and 2 by analyzing the general graphical representation of these beam functions in Figure 9. For each beam function to be optimal for its respective number of truck days ($\Delta^{opt.}_{C_N}$), the following statement must be valid:

$$\text{intersection of } Weekly\ costs^{opt. \Delta^{opt.}_{C^1_N}}(c) \text{ with } Weekly\ costs^{opt. \Delta^{opt.}_{C^2_N}}(c) \leq$$
$$\text{intersection of } Weekly\ costs^{opt. \Delta^{opt.}_{C^2_N}}(c) \text{ with } Weekly\ costs^{opt. \Delta^{opt.}_{C^3_N}}(c) \leq \ldots$$

Using the formulation used in Figure 9, we can therefore state that:

$C_P < C_N^{intersect.1,2} \leq C_N^{intersect.2,3} \leq ...$ .

In a scenario where the latter statement would not be true, some of the beam functions can never be optimal. If, for example, $C_N^{intersect.1,2} > C_N^{intersect.2,3}$, then $C_N^{intersect.1,3} < C_N^{intersect.1,2}$ because of the slope of the beam functions. In this case, and for the same reason, $Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^2}}(c)$ can never lead to an optimal solution because either $Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^1}}(c)$ or $Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^3}}(c)$ will always lie below $Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^2}}(c)$.

As Figure 9 gives a general representation (based on the properties of the intercepts, intersects and slopes) of the beam functions and the gable function, we make the following observations:

$\forall c \mid (C_P < c \leq C_N^{intersect.1,2})$:

$Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^1}}(c)$ results in the optimal weekly costs;

$\forall c \mid (C_N^{intersect.1,2} \leq c \leq C_N^{intersect.2,3})$:

$Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^2}}(c)$ results in the optimal weekly costs;

$\forall c \mid (C_N^{intersect.2,3} \leq c \leq \text{the next intersect})$:

$Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^3}}(c)$ results in the optimal weekly costs;

... .

Furthermore, as $Weekly\ costs^{opt.}_{\Delta^{opt.}_{C_N^x}}(c)$ is defined as the function that results in the optimal weekly costs when the cost of an N shift equals $C_N^x$ (i.e., when $c = C_N^x$), the following statement must be true based on the definition of the beam functions shown in Figure 9:

$$C_P < C_N^1 \leq C_N^{intersect.1,2} \leq C_N^2 \leq C_N^{intersect.2,3} \leq C_N^3 \leq ... .$$

Based on the properties of these beam functions with respect to the number of truck days and the required number of N shifts, it follows that:

$$\begin{cases} \forall\ C_N^1, C_N^2, C_N^3, ... \mid (C_P < C_N^1 \leq C_N^{intersect.1,2} \leq C_N^2 \leq C_N^{intersect.2,3} \leq C_N^3 \leq ...): \\ \qquad \Delta^{opt.}_{C_N^1} \leq \Delta^{opt.}_{C_N^2} \leq \Delta^{opt.}_{C_N^3} \leq ... \\ \forall\ C_N^1, C_N^2, C_N^3, ... \mid (C_P < C_N^1 \leq C_N^{intersect.1,2} \leq C_N^2 \leq C_N^{intersect.2,3} \leq C_N^3 \leq ...): \\ \qquad \rho^{opt.}_{\Delta^{opt.}_{C_N^1}} \geq \rho^{opt.}_{\Delta^{opt.}_{C_N^2}} \geq \rho^{opt.}_{\Delta^{opt.}_{C_N^3}} \geq ... \end{cases}$$

and that:

$$\begin{cases} \forall\ C_N^1, C_N^2, C_N^3, ... \mid (C_P < C_N^1 < C_N^{intersect.1,2} < C_N^2 < C_N^{intersect.2,3} < C_N^3 < ...): \\ \Delta_{C_N^1}^{opt.} < \Delta_{C_N^2}^{opt.} < \Delta_{C_N^3}^{opt.} < ... \\ \\ \forall\ C_N^1, C_N^2, C_N^3, ... \mid (C_P < C_N^1 < C_N^{intersect.1,2} < C_N^2 < C_N^{intersect.2,3} < C_N^3 < ...): \\ \rho_{\Delta_{C_N^1}^{opt.}}^{opt.} > \rho_{\Delta_{C_N^2}^{opt.}}^{opt.} > \rho_{\Delta_{C_N^3}^{opt.}}^{opt.} > ... \end{cases}$$

Hence, at each intersection of two consecutive (sorted according to the respective number of truck days) beam functions, the gable function features a breakpoint, indicating a change in the optimal number of truck days and the minimal required number of N shifts. Furthermore, the observations prove that for an increasing cost $c$, the optimal number of truck days increases and the minimal required number of N shifts decreases at each breakpoint. As none of the beam functions lie below the gable function, the following statement holds true with respect to each global optimum:

$$\begin{cases} \forall\ C_N^a, C_N^b, C_N^c, ... \mid (C_P < C_N^a < C_N^b < C_N^c < ...): \Delta_{C_N^a}^{opt.} \le \Delta_{C_N^b}^{opt.} \le \Delta_{C_N^c}^{opt.} \le ... \\ \forall\ C_N^a, C_N^b, C_N^c, ... \mid (C_P < C_N^a < C_N^b < C_N^c < ...): \rho_{\Delta_{C_N^a}^{opt.}}^{opt.} \ge \rho_{\Delta_{C_N^b}^{opt.}}^{opt.} \ge \rho_{\Delta_{C_N^c}^{opt.}}^{opt.} \ge ... \end{cases}$$

Note that we use the notation $C_N^a, C_N^b, C_N^c$ instead of $C_N^1, C_N^2, C_N^3$ to further emphasize that the latter statement is concerned with the gable function (represented by the solid line in Figure 9) instead of the beam functions. Hence, for increasing costs $c$, the optimal number of truck days will never decrease and the minimal required number of N shifts will never increase. □

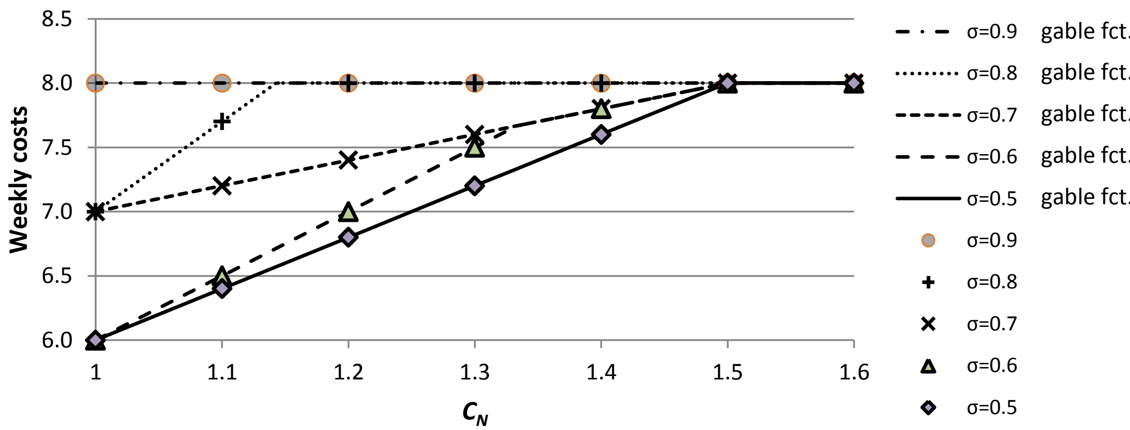
*5.1.2.2. Evaluating results based on the properties of an optimal solution*

Based on the previous lemmas and proven propositions, we can now evaluate the results in Table 2. Therefore, a gable function is constructed as a function of $C_N$ for each of the five different $\sigma$ values in our test cases (see Table 2). Figure 10 summarizes these five weekly cost functions that can be constructed for the results in Table 2 for each $\sigma$ value. The line graphs in Figure 10 represent the gable functions for the respective $\sigma$ values. The markers represent the weekly costs for each single solution in Table 2. To check the possibility that the obtained solutions in Table 2 are optimal, certain conditions should be met. Based on the lemmas and propositions stated before, the following four conditions should be met for each of the solutions within the same $\sigma$ case:

1. The number of truck days can never decrease for increasing $C_N$;

2. The minimal required number of N shifts can never increase for increasing $C_N$;
3. A breakpoint must be observed in the gable function for each change in the number of truck days; Moreover, the slope of the gable function should decrease at each breakpoint;
4. Each solution with the same number of truck days must contain the same number of N shifts. Hence, each solution should lie on its respective gable function.

Figure 10: Gable functions of the obtained results for different $\sigma$ cases




The results in column 4 in Table 2 satisfy condition 1 and condition 2 as for the obtained results the number of truck days will never decrease and the number of required N shifts will never increase with increasing $C_N$. The gable functions in Figure 10 also satisfy condition 3 as a breakpoint is observed in the gable function for each change in the number of truck days and the slope of each function will never increase at any breakpoint. Finally, condition 4 is also met since each solution (represented by a marker in Figure 10) lies on its respective gable function. Hence, the structure of the obtained solutions (for different values of $C_N$) satisfies all optimality conditions. This means that the structure of the obtained set of solutions resembles an optimal structure. Satisfying these four optimality conditions has two important implications. First, it indicates that we cannot reject the hypothesis of optimality of the individual solutions. It does, however, not prove the optimality of the individual solutions. Second, it proves that there is some consistency in the obtained solution quality for each single instance. In other words, our solution approach does not seem to arbitrarily perform better or worse for some problem instances as the structure of the obtained set of solutions resembles an optimal structure.

Hence, satisfying all optimality conditions outlined above does not mean that we can prove the optimality of the individual solutions. However, we can assume that this is very likely.

#### *5.1.2.3. Benchmarking*

While Section 5.1.2.2 only aims at confirming that our results satisfy the conditions required for an optimal solution, this section focuses on comparing the results with the results of a flexible schedule (see Section 4.5). Recall that in a fixed schedule, each container must be visited at least once a week. However, visiting a container does not necessarily mean that it will also be emptied. A container is only emptied when one of its compartments is filled for at least 40%. When a container is visited while it was not necessary (i.e., when none of its compartments is filled for at least the collection threshold), time was wasted. Based on sensor information, MUGO and MAGO containers can be defined. This allows to construct only the necessary collection routes, eliminating these useless trips. This means, however, that we have to abandon a fixed schedule and use a flexible schedule which can be different for each truck on each day in each week.

It is very likely that the obtained flexible schedule will result in fewer truck days since we only construct a collection route if necessary. However, the average weekly number of truck days of the flexible schedule cannot be seen as a theoretical lower bound (LB) since the rolling horizon approach does not take into account the long-term effects of the MUGO and MAGO selection decisions (see Section 4.5). Hence, it cannot be proven that the flexible schedule will always perform better than the fixed schedule. However, we argue that the results of the flexible schedule can be seen as a practical lower bound for the minimal effort (in terms of truck days) that is required to make sure that the glass overflow (glass placed next to the containers) is minimal. Johansson (2006), for example, concludes that with relative large systems (more than 100 containers), dynamic scheduling (i.e., constructing a flexible schedule) performs best. Johansson (2006) further concludes that the highest savings of this dynamic policy are achieved in environments with high demand fluctuation. Also Mes (2012) states that with seasonal patterns and huge random variations from day to day, a flexible schedule outperforms a fixed schedule. Both authors compare their flexible scheduling results to a fixed schedule and conclude that a flexible schedule can save on average 10% in working time or costs. As we also face a very large system (over 300 containers) and an environment with seasonal patterns and high glass disposal fluctuations, similar savings are expected.

Table 3 shows a comparison of the ME results with the results of the flexible schedule (LB) in terms of the minimal average required number of truck days. Again, we focus on the same four weeks as previously to produce the results and we make this comparison for the same five $\sigma$ cases. Columns 2 and 3 of Table 3 give the results for the scenario where we only consider N shifts. Columns 4 and 5 give the results for the scenario where we only consider P shifts. It is impossible to test the scenarios where we allow for a mix of P and N shifts since this is impossible within the framework of a flexible schedule. The procedure to obtain a flexible schedule (as outlined in Section 4.5) makes decisions on a daily basis and is therefore not designed to produce a good (or optimal) overall (over all days) mix of N and P shifts which satisfies the shift succession constraints. Therefore, only N or only P shifts are considered to analyze the LB.

As Table 3 shows, the LB is always smaller than the ME result. Note that the LB is an average calculated by dividing the total number of required truck days over the considered planning horizon by the number of weeks in the considered time horizon. Since each week can be different in a flexible schedule, the LB can therefore have a fractional value. This is of course impossible for the fixed schedule scenario. Hence, rounding the LB to the next integer also results in an even stronger (practical) lower bound for the fixed schedule scenario. Comparing the (rounded) LB and the ME results shows that the ME procedure produces the best possible solution for a fixed schedule, compared to a flexible schedule (at least for the cases where we consider only N or only P shifts). Along with the results in Section 5.1.2.2, this strongly suggests that our ME method is capable of finding good (even possibly optimal) solutions for the fixed schedule scenario.

Table 3: Comparison of the weekly average required number of truck days

| | Only N shifts | | Only P shifts | |
|---|---|---|---|---|
| Case | LB | ME | LB | ME |
| $\sigma = 0.9$ | 7.25 | 8.00 | 7.75 | 8.00 |
| $\sigma = 0.8$ | 6.50 | 7.00 | 7.75 | 8.00 |
| $\sigma = 0.7$ | 6.25 | 7.00 | 7.75 | 8.00 |
| $\sigma = 0.6$ | 5.75 | 6.00 | 7.75 | 8.00 |
| $\sigma = 0.5$ | 5.50 | 6.00 | 7.75 | 8.00 |

Note: LB (Lower Bound) refers to the average number of truck days resulting from the flexible schedule.
ME (Model Enhancement) refers to the average number of truck days resulting from the fixed schedule.

### *5.2. Application to a real-life problem*

In this section we present the results of the ME procedure for the entire year based on real-life data. The goal is to provide Fost Plus with an evaluation of the possibility to introduce N shifts in combination with the current P shifts. Furthermore, a flexible schedule is constructed to allow for an evaluation of the proposed results and the evaluation of installing sensors.

Table 4 presents the results of the ME procedure for the four most realistic $\sigma$ scenarios with $C_P = 1$. Solving the ME procedure for one year instead of 4 weeks for 100 iterations increases the required computation time from on average 10 minutes to on average 70 minutes. The results show that we always find a solution where $\theta_w^{simulation}$ is less than or equal to 7.5 hours. Just as for the results in Section 5.1.1, we can again observe that $\theta_w^{estimate}$ and $\theta_w^{simulation}$ lie very close to each other. Furthermore, the four conditions for an optimal solution as defined in Section 5.1.2.2 are also met, which indicates that the ME procedure consistently results in a good solution which could be optimal.

As in Section 5.1.2.3, Table 5 shows the results of the flexible schedule (LB) compared to the results of the ME procedure. Note that the LB is now the average required number of truck days over the entire year. Comparing the (rounded) (practical) LB and the ME results shows that the ME procedure produces the best possible solution for a fixed schedule compared to a flexible schedule (at least for those two cases). Along with the results in Table 4, this again suggests that our ME method is capable of finding good (even possibly optimal) solutions for the fixed schedule scenario.

Regardless of the optimality of the obtained results, the improvement with respect to the current schedule in use at the IA under study is at least as important in order to evaluate the quality of the proposed solutions. At this time, the IA under study uses a fixed schedule with only P shifts employing two full time drivers. Since one collection route is assigned to each driver on each day, 10 truck days are required every week in the current schedule. Even without the inclusion of an N shift, Table 4 shows that our model already results in a saving of at least three truck days per week. Apparently, there is quite some room for improvement regarding the current routes. Since a reduction of three truck days corresponds with a saving of 30% in labor costs, this can result in significant monetary savings. Table 4 shows that even more savings are possible when N shifts are used in combination with P shifts depending on the premium of an N shift and the value of $\sigma$. These results allow Fost Plus to evaluate different cost possibilities under different $\sigma$ assumptions.

To assure realistic results, we use the driving times provided by Google Maps. Hence, the driving times take into account speed limits, congestion, traffic lights and even the driving direction (A to B is different from B to A). To obtain the driving times from and to each of the more than 200 sites, the Google Maps API is used to automatically query the driving times. The Google Maps API also allows to obtain the driving directions for each collection route which can be graphically represented in the Graphical User Interface (GUI) designed for Fost Plus. Appendix 11 shows this GUI and all of its components. The GUI presents the user with the information of the fill level of each glass container (right), the composition of the collection routes for each truck (bottom) and of course a graphical representation of the collection route for each truck. It also allows the user to analyze a fixed or flexible schedule under different parameter settings such as driving time (e.g., highways or no highways), overtime, and drop-off rules. For instance, the user can analyze the possible effects of allowing a truck to go to the drop-off site at the end of the day without it being completely full.

Table 4: Real case results Fost Plus

| Case | $\sigma$ | $C_N$ | Truck days | Shifts | Weekly costs | Truck $w$ | $\theta_w^{estimate}$ | $\theta_w^{simulation}$ |
|---|---|---|---|---|---|---|---|---|
| 1 - 6 | 0.9 | 1.1 - 1.6 | 7 | 7P + 0N | 7.00 | 1 | 7.418 | 7.418 |
| | | | | | | 2 | 6.895 | 6.895 |
| 7 - 12 | 0.8 | 1.1 - 1.6 | 7 | 7P + 0N | 7.00 | 1 | 7.418 | 7.418 |
| | | | | | | 2 | 6.895 | 6.895 |
| 13 | 0.75 | 1.1 | 6 | 1P + 5N | 6.50 | 1 | 7.295 | 7.295 |
| | | | | | | 2 | 7.125 | 7.125 |
| 14 | 0.75 | 1.2 | 6 | 1P + 5N | 7.00 | 1 | 7.295 | 7.295 |
| | | | | | | 2 | 7.125 | 7.125 |
| 15 - 18 | 0.75 | 1.3 - 1.6 | 7 | 7P + 0N | 7.00 | 1 | 7.418 | 7.418 |
| | | | | | | 2 | 6.895 | 6.895 |
| 19 | 0.7 | 1.1 | 6 | 2P + 4N | 6.40 | 1 | 7.375 | 7.375 |
| | | | | | | 2 | 7.045 | 7.020 |
| 20 | 0.7 | 1.2 | 6 | 2P + 4N | 6.80 | 1 | 7.375 | 7.375 |
| | | | | | | 2 | 7.045 | 7.020 |
| 21 - 24 | 0.7 | 1.3 - 1.6 | 7 | 7P + 0N | 7.00 | 1 | 7.418 | 7.418 |
| | | | | | | 2 | 6.895 | 6.895 |
| | | | | | | Average: | 7.165 | 7.164 |
| | | | | | | St.dev.: | 0.245 | 0.246 |

Table 5: Comparison of the minimal average required number of truck days

| | Only N shifts | | Only P shifts | |
|---|---|---|---|---|
| Case | LB | ME | LB | ME |
| $\sigma = 0.9$ | 6.37 | 7.00 | 6.40 | 7.00 |
| $\sigma = 0.8$ | 6.10 | 7.00 | 6.40 | 7.00 |
| $\sigma = 0.75$ | 5.70 | 6.00 | 6.40 | 7.00 |
| $\sigma = 0.70$ | 5.34 | 6.00 | 6.40 | 7.00 |

Note: LB (Lower Bound) refers to the average number of truck days resulting from the flexible schedule. ME (Model Enhancement) refers to the average number of truck days resulting from the fixed schedule.

## 6. Conclusion and future research

In this paper we successfully showed how our model enhancement approach can be used to integrate the difficult vehicle routing problem (VRP) with the personnel planning problem at a company named Fost Plus. Our focus in this paper is on the glass collection process that is coordinated by this company. The glass collection is not performed by Fost Plus itself, but is decentralized to different IAs. These IAs collect the glass and charge Fost Plus with the collection costs. In this paper we analyze and compare two different possibilities that were proposed by Fost Plus in order to optimize the current collection process that is based on a fixed weekly workforce schedule.

First, we consider a second shift type (N shifts) on top of the single current shift type (P shifts). While the P shifts (containing the peak traffic hours) are cheaper than the N shifts (not containing the peak traffic hours), the driving times during the P shifts are on average higher than those during the N shifts. This difference creates a trade-off between higher costs and faster driving times which possibly results in a better workforce schedule with lower weekly labor costs. While this problem is related to the Periodic Vehicle Routing Problem (PVRP), our problem features several complicating elements that are absent in the standard PVRP such as shift scheduling and intermediate facilities. Furthermore, we assume that the load of each vehicle at the end of a day needs to be equal to the load of that vehicle at the start of the following day and that the daily fill rate of each compartment of each container is not constant over time. In order to deal with the increased complexity caused by these elements, we propose a simplified model as an approximation of the real model. We use a technique called model enhancement (ME) to ensure the quality and realism of this approximation by iteratively enhancing the assumptions of the simplified model based on simulation results. During each iteration, the simplified model is solved with a tabu search heuristic.

Second, Fost Plus wants to analyze the possible benefits of installing and using sensors in all glass containers in order to obtain a real-time view of the fill level of each container. Using this information, collection routes can be constructed on a daily basis in order to avoid wasting driving time to containers with sufficient capacity left (which is unavoidable in the case of a fixed weekly schedule). This means, however, that we have to abandon a fixed schedule and must use a flexible schedule that can be different for each truck on each day. Following a similar approach as Mes (2012) and Johansson (2006), we propose a rolling horizon technique where we define Must Go (MUGO) and May Go (MAGO) containers based on daily sensor information. Containers that should be emptied today in order to avoid overflow are referred to as MUGOs. MAGO containers are selected based on their level of urgency. In order to test the rolling horizon procedure, a simulation model is used.

Both solution approaches are applied to several test cases and to the real-life data from a particular IA. In order to evaluate the performance of the ME model, both the feasibility and the solution quality of the obtained results are analyzed. The feasibility of the obtained result is an important performance indicator of the ME approach as it shows how good the simplified model approximates the real model. The results show that each obtained solution (for all test cases as well as the real-life case) is feasible in both the simplified and the real model. Hence, the enhanced assumptions of the simplified model are a good approximation of reality. Analyzing the quality of the objective values (i.e., the total weekly labor costs) of the obtained ME results is less straightforward.

First, a rather unconventional strategy is employed to prove that the obtained set of solutions satisfies the properties of a set of optimal solutions. As the results indeed meet all optimality conditions for all test cases, we first explained that the hypothesis of optimality of the individual solutions cannot be rejected. That does, however, not prove the optimality of the individual solutions. Second, we showed that our solution approach can deliver consistent results and does not seem to arbitrarily perform better or worse for some problem instances. Hence, while we cannot prove that our solution approach always results in the optimal solution, we can assume that this is very likely.

Second, the quality of the obtained ME results is evaluated based on a practical lower bound (LB). This lower bound is calculated based on the obtained results for a flexible schedule. In the flexible scheduling model, useless trips are eliminated since only the most urgent containers are visited. Therefore, the obtained simulation results for a flexible schedule can be seen as a good lower bound for the minimal effort that is required for glass collection in order to minimize glass overflow (glass placed next to the containers). As the ME results are equal to the rounded LB results for the tested scenarios, this strongly suggests that the proposed ME method is capable of finding good (even possibly optimal) solutions.

Regardless of the optimality of the obtained results for the test cases, the improvement with respect to the schedule used in reality is at least as important in order to evaluate the quality of the proposed techniques. We show that even without the inclusion of an N shift, the ME model results in significant monetary savings (30%). Even more savings are possible when N shifts are used in combination with P shifts depending on the cost premium of an N shift. Furthermore, the results presented in this paper also allow Fost Plus to evaluate the costs and benefits of installing sensors in each container. This analysis goes, however, beyond the scope of this paper.

Finally, we propose some interesting topics for future research. In this paper we assume to know the exact fill level of the containers based on historical fill rate data. However, some stochasticity can be expected in the fill rate. Hence, incorporating a stochastic fill rate can contribute to the construction of a more realistic and applicable model. Furthermore, stochasticity can also be introduced in the driving times, collection times, loading and unloading times, etc. However, while stochasticity is an interesting additional element, we rather prefer to emphasize the lack of research related to the combination of personnel and vehicle shift scheduling in the waste management literature. This lack was already pointed out by Ernst et al. in 2004 and still exists today according to the literature review of Ghiani et al. (2014).

## 7. References


F. Alonso, M. J. Alvarez and J. E. Beasley. A tabu search algorithm for the periodic vehicle routing problem with vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, 59:963–976, 2008.

H. Andersson, A. Hoff, M. Christiansen, G. Hasle and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

E. Angelelli and M. G. Speranza. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137(2):233–247, 2002.

B. Bachelet and L. Yon. Model enhancement: Improving theoretical optimization with simulation. *Simulation Modelling Practice and Theory*, 15(6):703 – 715, 2007.

J. Baudach, A. Chmielewski and U. Clausen. Integrated vehicle routing and crew scheduling in waste management (part II). In C. Barnhart, U. Clausen, U. Lauther and R. H. Möhring, editors, *Models and Algorithms for Optimization in Logistics*, number 09261 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

A. Campbell, L. Clarke, A. Kleywegt and M. Savelsbergh. The inventory routing problem. In *Fleet Management and Logistics*, pages 95–113. Kluwer Academic Publishers, 1998.

T. Carić and H. Gold. Enhancing solution similarity in multi-objective vehicle routing problems with different demand periods. In T. Murata and R. Itai, editors, *Vehicle Routing Problem*, chapter 7. In-Teh Publishers, 2008.

L. C. Coelho, J.-F. Cordeau and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

S. Coene, A. Arnout and F. C. R. Spieksma. The periodic vehicle routing problem: A case study. *Journal of the Operational Research Society*, pages 1–10, 2010.

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

P. M. Francis, K. R. Smilowitz and M. Tzur. The period vehicle routing problem and its extensions. In B. Golden, S. Raghavan and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 73–102. Springer US, 2008.

G. Ghiani, D. Laganá, E. Manni, R. Musmanno and D. Vigo. Operations research in solid waste management: A survey of strategic and tactical issues. *Computers & Operations Research*, 44:22–32, 2014.

G. Ghiani, E. Guerriero, A. Manni, E. Manni and A. Potenza. Simultaneous personnel and vehicle shift scheduling in the waste management sector. *Waste Management*, 33 (7):1589–1594, 2013.

F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Boston, 1997.

B. Golden, A. Assad and R. Dahl. Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems*, 7(2):181–190, 1984.

R. Hansmann and U. Zimmermann. Integrated vehicle routing and crew scheduling in waste management (part i). In C. Barnhart, U. Clausen, U. Lauther and R. H. Mohring, editors, *Models and Algorithms for Optimization in Logistics.* Number 09261 in Dagstuhl Seminar Proceedings, 2009.

P. Jaillet, L. Huang, J. F. Bard and M. Dror. A rolling horizon framework for the inventory routing problem. *Research paper, University of Texas*, 1:1–32, 1997.

O. M. Johansson. The effect of dynamic scheduling and routing in a solid waste management system. *Waste Management*, 26(8):875–885, 2006.

B.-I. Kim, S. Kim and S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642, 2006.

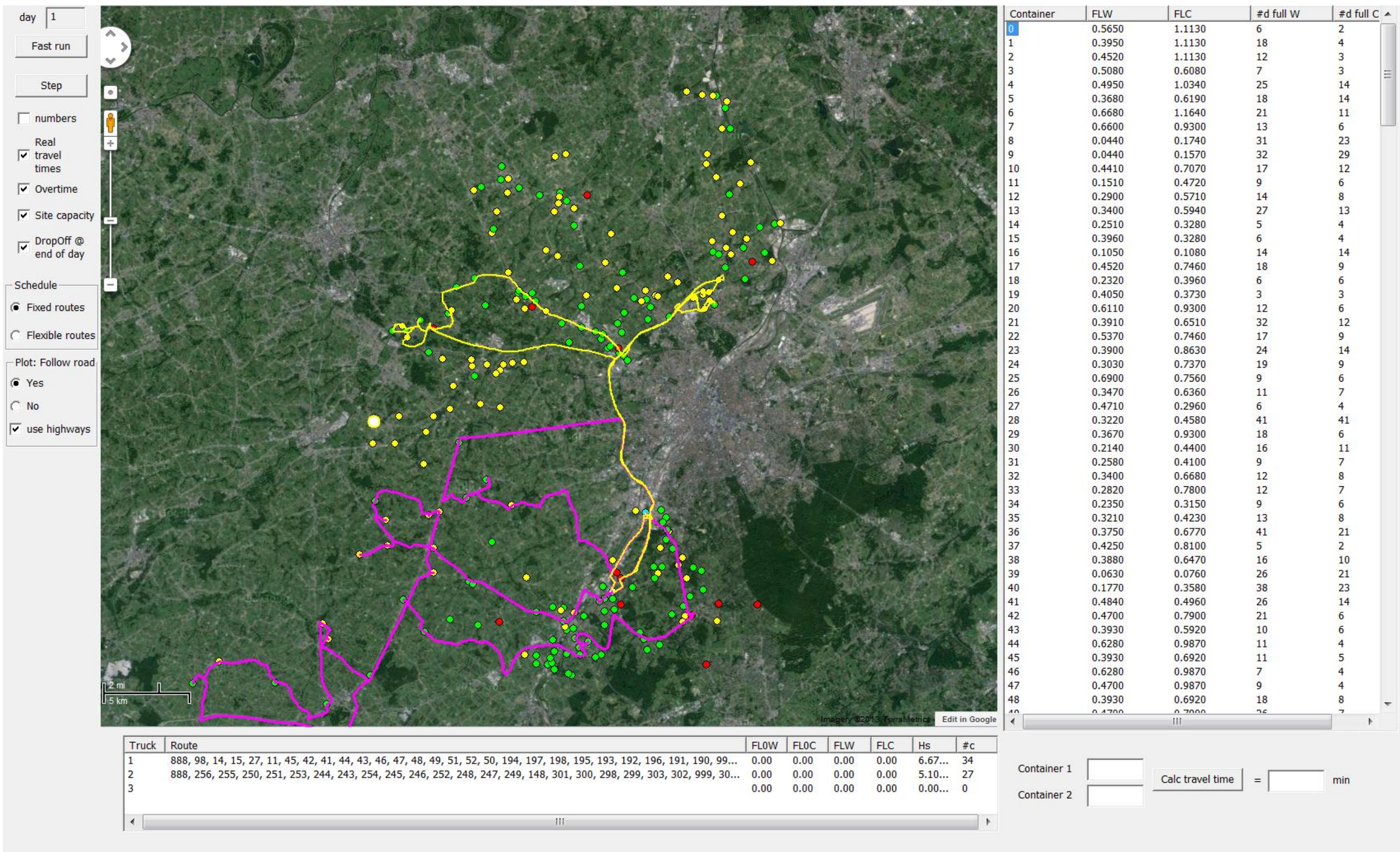
G. F. List, B. Wood, M. A. Turnquist, L. K. Nozick, D. A. Jones and C. R. Lawton. Logistics planning under uncertainty for disposition of radioactive wastes. *Computers & Operations Research*, 33(3):701–723, 2006.

M. Mes. Using simulation to assess the opportunities of dynamic waste collection. In S. Bangsow, editor, *Use Cases of Discrete Event Simulation*, pages 277–307. Springer Berlin Heidelberg, 2012.

E. Shamshiry, B. Nadi and A. R. Mahmud. Optimization of municipal solid waste management. *In: Proceedings of the 2010 international conference on biology, environment and chemistry*, 1:19–21, 2011.

## 8. Appendix

Figure 11: Graphical User Interface



**FACULTY OF ECONOMICS AND BUSINESS**
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be