Strategic Time Slot Management:
A Priori Routing for Online Grocery Retailing

Thomas R. Visser*[1] and Martin W.P. Savelsbergh[2]

[1] *Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands*
[2] *H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, USA*

Econometric Institute Report Series – EI2019-04

Abstract

Time slot management refers to the design and control of the delivery time slots offered to customers during the online ordering process. Strategic time slot management is an innovative variant in which only a single time slot is offered each day of the week and a priori delivery routes are used to guide time slot availability. Strategic time slot management simplifies time slot control and fulfillment center operations. We propose a 2-stage stochastic programming formulation for the design of a priori delivery routes and time slot assignments and a sample average approximation algorithm for its solution. An efficient dynamic program is developed for calculating the expected revenue of an a priori route. An extensive computational study demonstrate the efficacy of the proposed approach and provides insights in to the benefits of strategic time slot management.

Keywords: online grocery retailing, home delivery, time slot management, a priori routing, dynamic programming, sample average approximation

1 Introduction

This paper studies a novel variant of Time Slot Management inspired by its application in online grocery retailing. Online retailing continue to grow, and consumers, but also small businesses, purchase more and more products online. Many of these products require *attended* delivery, i.e., delivery can only take place when the consumer is present. Examples of such products include, furniture, white goods, and groceries. Delivery failures, for instance when the consumer is not at home, are costly, because products have to be returned and stored, and deliveries have to be re-scheduled. In case of groceries, the cost of a delivery failure may be even higher, as most grocery products are perishable and re-delivery of the (same) product is not possible. To minimize the chance of delivery failures, online retailers typically allow their customers to choose a delivery time slot. These time slots can range from 1 to 6 hours, and can have different delivery fees. The retailer agrees to deliver any ordered products in the time slot chosen by the customer. While offering delivery time slots improves customer service and reduces the chance of delivery failures, its implementation is challenging as time slot management needs to balance the benefit of accepting orders and the cost of delivering accepted orders.

*Corresponding author. Email: t.r.visser@ese.eur.nl

In this paper, we focus on time slot management for online grocery retailers. During the ordering process, customers log-in on a grocery retailer's website, fill their order basket, and place their order by selecting an available (delivery) time slot. Customers can select from among a set of available time slots in the few days (or even weeks). At the cut-off time for deliveries on a particular day, usually 12 to 18 hours before the departure of the delivery vehicles, vehicle routes and schedules for delivery of the placed orders are generated, and, afterwards, picking of the orders can commence in the fulfillment center. Then, if no issues arise during the execution of the delivery routes, customers receive their placed orders during their selected time slot. Online grocery retailers face varying daily demand (i.e., the number and the location of customers placing orders on any given day), but, because of the recurring nature of grocery purchases, also observe recurring patterns. Customers often have a favorite time slots and delivery days, and often order with some regularity, e.g., every week or other week.

Time Slot Management (TSM), as the name suggests, refers to the methods employed to manage the availability of time slots during the ordering process. TSM methods can be divided in two classes: static and dynamic (Agatz et al., 2013). Static Time Slot Management, sometimes called static slotting, partitions the set of customer locations into geographical regions and limits the number of customer orders that can be placed in a particular region (Agatz et al., 2011; Hernandez et al., 2017; Bruck et al., 2018); sometimes pricing of deliveries is also considered (Klein et al., 2017). The limit on the number of customer orders that can be placed in a region is determined upfront, and is based expected demand and expected delivery capacity. With static TSM the management of time slots during the ordering process is straightforward, but determining the regions and the limit for each region, given unknown varying demand is challenging. Moreover, the number of vehicles required to deliver orders can vary from day to day. After the cut-off time, vehicle routes and schedules are generated, and only upon completion of this process is the number of vehicles required known, which may cause operational challenges. Furthermore, picking of customer orders at the fulfillment center can only start once the vehicle routes and schedules are generated, because orders delivered in the same route are typically picked together. Dynamic Time Slot Management, sometimes called dynamic slotting, maintains partial delivery routes and schedules during the ordering process and uses these to guide decisions on the availability of time slots (Campbell and Savelsbergh, 2005; Ehmke and Campbell, 2014; Köhler et al., 2019) – sometimes pricing of deliveries is also considered (Campbell and Savelsbergh, 2006; Cleophas and Ehmke, 2014; Yang et al., 2016). By maintaining partial vehicle routes and schedules it becomes easier to adjust to variations in demand and it allows control of the number of vehicles required to deliver orders. After the cut-off time, the vehicle routes and schedules are re-optimized. Again, picking of customer orders at the fulfillment center can only start once the vehicle routes and schedules have been finalized. Clearly, the management of time slots during the ordering process is much more involved than with static TSM, especially if detailed and accurate vehicle routes and schedules are maintained, e.g., accounting for time-dependent travel times and driver breaks (Ehmke and Campbell, 2014). Therefore, it is not surprising, that online grocery retailers are continuing to look for business models that simplify time slot management without reducing customer service and increasing delivery costs.

In this paper, we investigate a novel variant of TSM which we call *Strategic Time Slot Management* (STSM). It is motivated by the current practice of online grocery retailer Picnic in The Netherlands (`https://picnic.app/nl/`). Picnic raised a record €100 million in venture funding after just 1.5 years of operations (Sterling, 2018). Picnic operates in all major city centers of the Netherlands and delivers customer orders using small electric delivery vehicles with limited range and capacity. Picnic does not charge a delivery fee to their customers. Picnic offers only a single 1-hour time slot each day, but it varies over the days of the week (e.g., the same single 1-hour time slot is available to a customer every Monday, but a different time slots is available every Tuesday). This suggests that Picnic designs, for each day of the week, *a priori* routes, covering all customer locations, and assigns time slots to the customer locations visited on these a priori routes *before* the order placement process. Then, during the ordering process, the availability of time slots is managed using these a priori routes. Customers that place an order in a time slot available to them are inserted in a delivery route associated with the a priori route. A time slot for a customer location in an a priori route is available as long as the location can be feasibly visited given the orders that have already been placed; otherwise, the time slot is no longer offered. Managing time slots in this way is easy if delivery routes "follow" the a priori routes, skipping locations that can no longer be visited during the assigned time slot.

Employing STSM implies that fewer time slots are available to customers, but it allows for more cost-effective operations and offering free delivery to customers. Picnic's success indicates that is a winning business proposition. Because a priori routes are designed to be operated for a period of time, it is easier to incorporate knowledge of the order patterns of customers, which can have advantages especially in urban areas with large number of densely distributed customers. Maybe more importantly, no (re)optimization of vehicle routes and schedules is required after the cut-off time, which means that a later cut-off time can be offered to customers. And not only that, picking at the fulfillment center can usually start *before* the cut-off time, as soon as (the initial part of) the delivery routes are known because of orders that have already been placed. This not only improves efficiency of fulfillment center operations, it, again, allows the cut-off time to be pushed later. Finally, the use of a priori routes induces delivery route consistency, which implies that drivers familiarize themselves with their delivery routes, which is helpful in densely populated city centers and improves customer service (Kovacs et al., 2014).

Clearly, the design of the a priori routes and time slot assignments is critical to the success of STSM. This design problem shares some characteristics with the stochastic vehicle routing problems studied in the literature (see Gendreau et al. (2014); Oyola et al. (2017, 2018) for recent surveys on stochastic VRPs). However, much of the research on stochastic VRPs has focused on uncertainty regarding the *size* of demand. In the context of STSM, it is not the size of demand that matters most, because vehicle capacity is rarely constraining, but rather the uncertainty regarding the placement of orders (i.e., stochastic customer presence), because it is the time available to make deliveries that is constraining. The concept of, or use of the term, a priori routes is also quite common (see Campbell and Thomas (2008a) for a survey on a priori routing). In all the considered settings, there is a design phase, in which a priori routes have to be determined, and an execution phase, in which the uncertain quantity is revealed and the a priori routes have to be executed, given a set of *recourse actions* or *penalities* to handle situations in which certain constraints are violated. Typically, the a priori routes are constructed using probabilistic information about the uncertain quantity and the set of recourse actions or penalties. A common approach is to formulate the (design) problem as a two-stage stochastic program.

For stochastic customer presence, most research has focused on the Probabilistic Traveling Salesman Problem (PTSP) and its variants (see for instance Jaillet, 1988; Campbell and Thomas, 2008b; Voccia et al., 2013; Angelelli et al., 2017). The basic problem seeks an a priori route with minimum expected cost. Campbell and Thomas (2008b) consider a variant in which customers have deadlines and Voccia et al. (2013) consider a variant in which customers have time windows. Erera et al. (2009) study a related a priori routing problem in which customers with time windows are assigned to both a primary and a backup route, and once customer presence is revealed, recourse actions can move the customers from the primary to a backup route to improve costs or to recover feasibility. The setting we consider is somewhat different, in the sense that there are no recourse actions or penalties. Time slot management ensures the feasibility of the delivery routes at the end of the ordering processing, i.e., customers are revealed sequentially and are only shown time slots for which delivery is still feasible. To the best of our knowledge, such a setting, i.e., an ordering process with time slot management, has not yet been considered in the a priori routing literature.

The planning problem at the heart of STSM seeks not only a set of a priori routes, but also the assignment of a time slot to each of the locations visited on the a priori routes. The assignment of time slots to vehicle routes has been investigated in other contexts. In the Time Window Assignment Vehicle Routing Problem (TWAVRP) (Spliet and Gabor, 2015; Spliet and Desaulniers, 2015) time slots have to be assigned to customers before their demand is known, and vehicle routes are generated only when demand is revealed. Spliet and Gabor (2015) formulate the problem as a two-stage stochastic program and develop a branch-and-cut-and-price approach. The Consistent Vehicle Routing Problem (Groër et al., 2009), and its singe-vehicle variants (Subramanyam and Gounaris (2016), Subramanyam and Gounaris (2018)), seek vehicle routes that are to be executed on multiple days, with known, but varying customer presence on each of the days, and that minimize the difference in the time of service of a customer on the different days. In these settings all customer demand needs to be served, whereas in our setting time slot management during the ordering process may result in some realized customer demand not being served. Furthermore, in our setting a priori routing and time slot assignment are integrated into a single planning problem, rather than treating these aspects separately.

The contributions of this paper are as follows:

- We introduce the concept of strategic TSM, a novel variant of TSM inspired by operations of a Dutch online grocery retailer.
- We derive a number of properties of the single-vehicle variant of the strategic TSM planning problem, and use these observations to develop an efficient dynamic programming algorithm for exactly calculating the expected revenue of an a priori route.
- We present a two-stage stochastic programming formulation for the single-vehicle variant of the strategic TSM planning problem, and develop a Monte-Carlo Sample Average Approximation (SAA) Method (Kleywegt et al., 2002). Time slot management during the ordering process, i.e., when orders are sequentially revealed, leads to an "evaluation" recourse, which has not been seen in the context of SAA.
- We propose a number of heuristic methods for the single-vehicle variant of the strategic TSM planning problem, which balance quality and solution time.
- We compare solution approaches on instances with up to 12 customer locations on an a priori route,

which implies around $1.3 \cdot 10^9$ possible customer arrival scenarios. Moreover, we investigate the impact of different time slot configurations (time slot width and whether or not time slots overlap) and the relation between the duration of an a priori route and the target duration of the delivery route (which impacts time slot management).

The paper is structured as follows. In Section 2, we introduce the Strategic TSM problem, focusing specifically on the "simpler" single-vehicle case, and we present some observations that help guide the design of solution approaches. In Section 3, we discuss algorithms to exactly evaluate the expected revenue of an a priori route, which is a core component of our solution approaches. In Section 4, we formulate the problem as a two-stage stochastic program and propose an SAA method for its solution. In Section 5, we present heuristic solution approaches seeking to balance quality and solution time. In Section 6, we discuss the results of an extensive computational study. Finally, in Section 7, we present concluding remarks and suggest future research directions.

2 Problem Description

We consider a retailer that offers its online customers a small number of time slots during which a delivery can take place. The retailer has a fleet of identical vehicles to make deliveries. Each vehicle starts and ends its delivery route at the retailer's fulfillment center. For each of its customers, the retailer knows the delivery location, the order size, the revenue, and the order placement probability. Observe that the only stochastic feature in this setting is whether or not a customer places an order. In practice, a customer's order size and revenue are likely to be stochastic as well.

Customers can place an order up to a cut-off time, some hours before delivery will take place. We assume that the likelihood that a customer places an order is independent of the delivery time slots offered and is not correlated to the order placement of other customers. When placing an order, a customer must select a delivery time slot during which delivery will take place at his delivery location. A vehicle that arrives early at a delivery location must wait.

The retailer seeks to *design* a set of delivery routes, such that each customer, i.e., its delivery location, is visited on at least one of the routes, and associated time slots, one for each location visited, so as to maximize the *expected* revenue.

We assume that the set of possible time slots that can be assigned to a customer location has already been decided. The time slots may overlap, but they all have the same width, and they cover the entire planning horizon. The subset of possible time slots for a delivery location contains those time slots that are feasible for that location, i.e., that overlap with the time period defined by the earliest time a vehicle can reach the location and the latest time a vehicle can depart the location to return to the fulfillment center before the end of the planning horizon.

As will become evident soon, even solving the special case in which the retailer has only a single vehicle with infinite capacity and assigns only a single time slot to each delivery location is surprisingly challenging and gives rise to insightful observations. For the remainder, therefore, we focus on this special case, leaving the general case for future research.

2.1 The Single-Vehicle Case

The problem is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, with $\mathcal{V} = \mathcal{V}_\text{c} \cup \{o, d\}$ the set of vertices, where $\mathcal{V}_\text{c} = \{1, 2, \ldots, n\}$ is the set of customer delivery locations and where, for convenience, we represent the fulfillment center with a start and an end node, o and d, respectively, to be able to distinguish the departure and return of the vehicle, and with $\mathcal{A}$ the set of (directed) arcs connecting the nodes. We let $t_{ij} \geq 0$ denote the travel time associated with arc $(i, j) \in \mathcal{A}$. We assume that service times are included in the travel times, and that travel times satisfy the triangle inequality. The retailer has a single vehicle with unlimited capacity to make deliveries, which reflects that it is time rather than capacity that restricts the delivery route. When a time slot $s = [a_s, b_s]$ is assigned to a location in the delivery route, the earliest time a delivery can be made at that location is a_s and the latest time a delivery can be made at that location is b_s. A vehicle arriving early must wait at the location. A set $\mathcal{T}$ of possible time slots to be assigned to delivery locations is given. The time slots in $\mathcal{T}$ may overlap, but we assume their width is equal, and they cover the entire planning horizon $[0, T]$, with T the planning horizon. The set of possible time slots $\mathcal{T}_i \subset \mathcal{T}$ for location i contains the time slots which overlap with the period defined by the earliest time a vehicle reach that location, i.e., $t_{o,i}$, and the latest time a vehicle has to depart from that location (to return to the fulfillment center before the end of the planning horizon), i.e., $T - t_{i,d}$. The fulfillment center has time window $[a_o, b_o] = [a_d, b_d] = [0, T]$. We identify the set of customers $\mathcal{C}$ with their delivery locations, i.e., $\mathcal{V}_\text{c}$ (and use these interchangeably from now on). Each customer $c \in \mathcal{C}$ has an order placement probability $p_c \in (0, 1]$, and, when served, results in a revenue r_c for the retailer. We assume that order placement probabilities are iid and independent of the time slot assigned to the delivery location.

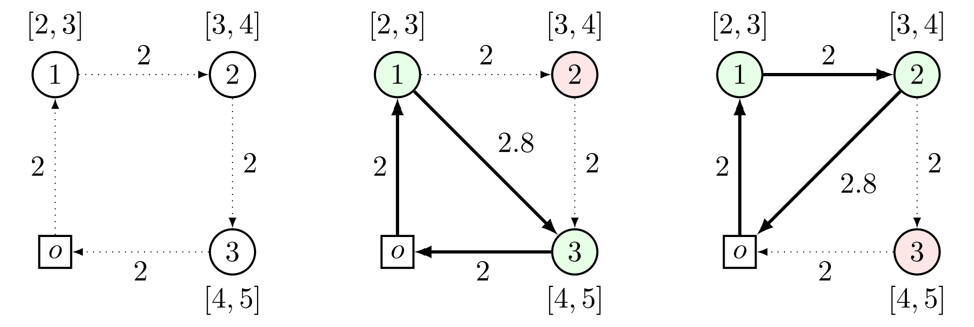
The retailer seeks to design an *a priori* delivery route, visiting all delivery locations, and associated time slots, one for each location, so as to maximize the *expected* revenue.

Let Ω be the set of all possible scenarios of order placements. A single scenario $\omega \in \Omega$ can be described by a sequence of delivery locations, representing which customers have placed an order and in what sequence – the exact times of the order placements are not important. Furthermore, we assume that each permutation of customer placements is equally likely, meaning that there is no dependence between customers and their position in the sequence. Each customer is equally likely appear early in the sequence as to appear late in the sequence. (Note that when the order placement probabilities are equal, i.e., $p_c = p$ for $c \in \mathcal{C}$, all possible scenarios are equally likely – given the iid assumption.)

The revenue for a scenario $\omega \in \Omega$ is determined as follows. During the order placement phase, an arriving order is inserted in the *actual* delivery route, i.e., the delivery route to be executed after the cut-off time, based on the delivery location's position in the a priori route. That is, the delivery location is inserted in the actual delivery route after the delivery locations of orders placed earlier and that precede it in the a priori route, and before the delivery locations of orders placed earlier and that succeed it in the a priori route. After the insertion of an order, any delivery location that has become time infeasible, i.e., for which it is no longer possible to make a delivery during its assigned time slot, is removed, and orders for these locations will be skipped from that point on. After all orders in ω have been processed, i.e., have either been inserted or skipped, the revenue of the scenario is simply the sum of the revenues of the orders that have been inserted in the actual delivery route. The expected revenue for an a priori route is the sum of the revenues of all possible scenarios for that a priori route weighted by the probability of occurrence of the scenarios.

Observe that (to keep operations simple) the delivery locations in the actual delivery route are visited in the same order as in the a priori route.

2.1.1 Small Example



(a) An a priori route with assigned time slots.

(b) Delivery route for customer realization $(3, 1, 2)$. After customers 3 and 1 place an order, customer 2 cannot by infeasibility.

(c) Delivery route for customer realization $(2, 1, 3)$. Now customer 3 cannot place an order.

Figure 1: Small example of Strategic Time Slot Management.

To illustrate the single-vehicle problem, we now present a small example. Let us consider three customer locations $\mathcal{V}_{\text{c}} = \{1, 2, 3\}$ with coordinates $\{(2, 0), (2, 2), (0, 2)\}$, respectively, and a fulfillment center located at coordinates $(0, 0)$ with a planning horizon $[0, T] = [0, 7]$. Each customer location has equal order probability $p = \frac{1}{2}$ and equal revenue $r = 1$. The set of possible time slots is given by $\mathcal{T} = \{[0, 1], [1, 2], \ldots, [6, 7]\}$. The travel times are given by the Euclidean distances.

A solution to the single-vehicle problem consist of an a priori route and a time slot assignment. Let us consider the a priori route $\rho = (o, 1, 2, 3, d)$ and time slot assignment $\{[2, 3], [3, 4], [4, 5]\}$ for locations $\{1, 2, 3\}$, respectively. This solution is shown in Figure 1(a).

Suppose now, with this design in place, that customer 3, then customer 1, and then customer 2, (each in $\mathcal{C}$) seek to place an order. This corresponds to scenario $\omega = (3, 1, 2) \in \Omega$. After placement of customer 3, the delivery route will be $(o, 3, d)$, and after placement of customer 1, the delivery route will be $(o, 1, 3, d)$, since each customer is inserted at their corresponding position in the a priori route and both can be inserted without violating time slots or the fulfillment center time window. This is shown in Figure 1(b). Notice that now, insertion of customer 2 at its position in the a priori route, resulting in delivery route $(o, 1, 2, 3, d)$, will violate the fulfillment center time window. This latter route requires a total of 8 time units while the fulfillment center time window is $[0, 7]$. Therefore, the time slot for customer 2 will be removed, and this customer cannot place an order. The resulting delivery route has revenue 2. In Figure 1(c), we now consider scenario $\omega = (2, 1, 3) \in \Omega$. Now customer 3 cannot place an order, resulting in a different delivery route $(o, 1, 2, d)$ with revenue 2. Notice that not only which customers arrive, but also the sequence in which they arrive determines the eventual delivery route. We have to take this into account when designing the a priori route and time slot assignment.

2.1.2 Observations

We highlight some interesting (and possibly unexpected) observations regarding this single-vehicle design problem.

Observation 1. *When the planning horizon T is greater than or equal to T^{TSP}, the minimum duration tour visiting all customer locations, the single-vehicle problem is trivial. An optimal a priori route is a minimum duration tour and an optimal time slot assignment is one in which the time slot assigned to a customer location contains the arrival time of the tour at that location. The expected revenue is $\sum_{i \in \mathcal{V}_c} p_i r_i$.*

It is natural to think that by increasing the planning horizon T, i.e., the time available for the delivery route, the expected revenue will increase (or at least not decrease). However, it turns out that this is not always true.

Observation 2. *Increasing the planning horizon T, i.e., the delivery route time limit, for a given a prior route may decrease the expected revenue.*

Figure 2 shows a simple example of this with three customers. Customers have probability $p = \frac{1}{2}$ and coordinates $\{(0,2),(0,1),(0,-1)\}$, respectively, and the travel times are equal to the Euclidean distances. Customer 1 has revenue 10, and customers 2 and 3 each have revenue 1. Suppose all customers are assigned

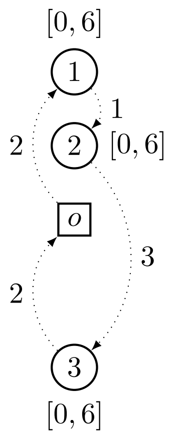
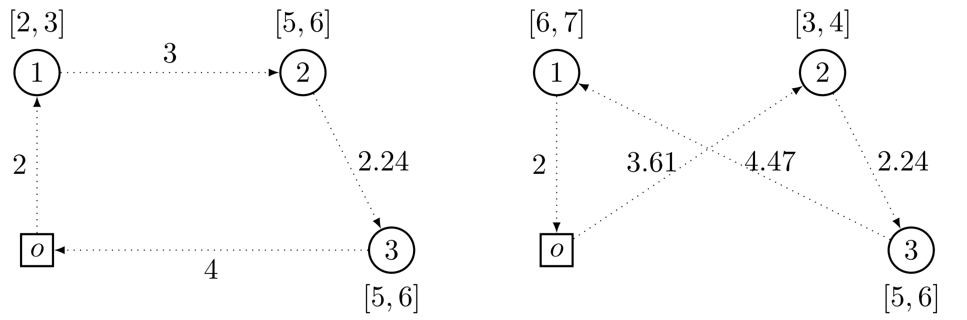


Figure 2: A design for a small example with three customers for which increasing the planning horizon from $6 - \epsilon$ to 6 results in a decrease in expected revenue.

time slot $[0, 6]$. When increasing the planning horizon from $6 - \epsilon$ to 6, the expected revenue (associated with the optimal solution) decreases from 4.646 to 4.625. The reason is that when the planning horizon is $6 - \epsilon$, the revenue for scenario $\omega = (2, 3, 1)$ is 11 (customer 3 cannot place order), but when the planning horizon is 6, the revenue reduces to 2 (customer 3 can place order, but customer 1 not). This decrease in revenue in this scenario is more than the increase in revenue in the other scenarios, and therefore the expected revenue decreases. We observe this effect also in larger instances, even when customers have equal revenue $r = 1$, equal probability $p = \frac{1}{2}$, and the minimum duration tour is taken as the a priori route.

It is natural to think that the a priori route has to be a minimum duration tour visiting all customer locations. However, it turns out that this is not always true.



(a) An optimal solution with expected revenue 1.083 when a priori route is fixed to the TSP optimal tour.

(b) An optimal solution with expected revenue 1.25, without fixing the a priori route.

Figure 3: Two designs for a small example with three customers and planning horizon $T = 9.85$.

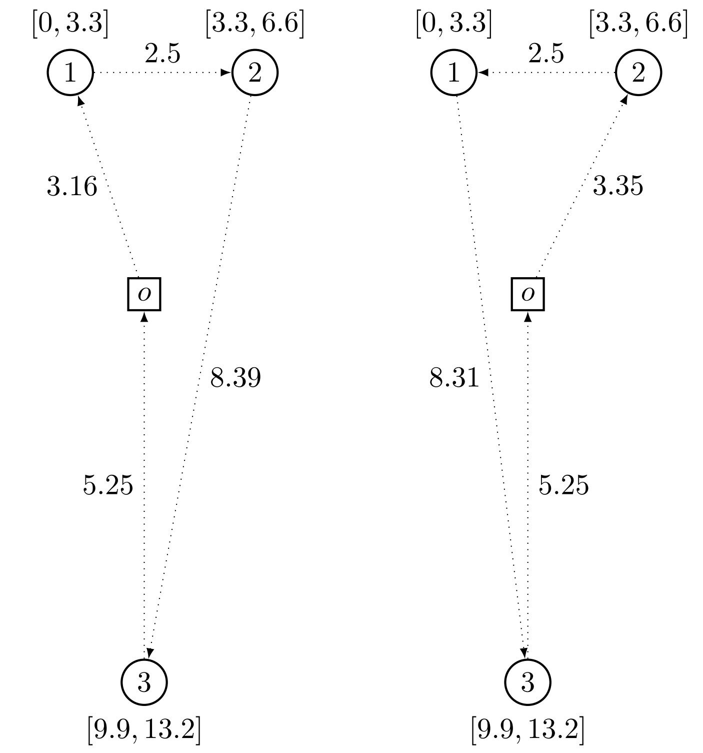
Observation 3. *The optimal a priori route is not always a minimum duration tour.*

We can show this with an example, depicted in Figure 3, with three customers, each with probability $p = \frac{1}{2}$ and revenue $r = 1$, planning horizon $T = 9.85$, and possible time slots $\mathcal{T} = \{[0,1],[1,2],\ldots\}$. Customers have coordinates $\{(2,0),(2,3),(0,4)\}$, respectively, and the travel times are equal to the Euclidean distances. Figure 3(a) shows the optimal design when the a priori route is forced to be a minimum duration tour (the design decision involves the direction in which the minimum duration tour is traversed in the time slot assignment). The expected revenue is 1.083. In this design, customers $\{1,2\}$ can be served together, but customers $\{2,3\}$ cannot be served together due to their assigned time slots, and customers $\{1,3\}$ cannot be served together due to the available time in planning horizon. Figure 3(b) shows the optimal design when the a priori route is not forced to the minimum duration tour. The expected revenue is 1.25. In this design, customers $\{1,2\}$ and customers $\{2,3\}$ can be served together, while only customers $\{1,3\}$ cannot be served together.

It is natural to think that the time slots along the a priori route should be *ascending* in both start- and end times, i.e., the start- and end times of the time slots increase along the a priori route. However, it turns out that this is not always true.

Observation 4. *In an optimal design, the time slots assigned to the customer locations are not always ascending along the a priori route.*

We can show this with an example, depicted in Figure 4, of three customers with equal probability $p = \frac{1}{2}$ and revenue of 1 for customers 1 and 2 and revenue of 5 for customer 3. Customers have coordinates $\{(-1,3),(1.5,3),(0,-5.25)\}$, respectively, and the travel times are, again, equal to the Euclidean distances. The set of possible time slots is $\mathcal{T} = \{[0,3.3],[3.3,6.6],\ldots\}$ and the planning horizon is $T = 19.29$. Figure 4(a) shows the optimal design when time slots are forced to be ascending along the a priori route, which results in an expected revenue of 3.208. Figure 4(b) shows the optimal design when there are no restrictions on the time slot assignments, which results in an expected revenue of 3.250. Note that time slot of customer 2 is $[3.3, 6.6]$ and that the time slot of the next customer on the a prior route, i.e., customer 1, is $[0, 3.3]$. The increase in expected revenue is due to scenarios $\omega = \{1,2,3\}$ and $\omega = \{2,1,3\}$. When the time slots are forced to be ascending, the resulting design allows customers 1 and 2 to place their



(a) Time slots restricted to be ascending; expected Revenue: 3.208.

(b) Time slots not restricted; expected Revenue: 3.250.

Figure 4: Two designs for a small example with 3 customers and planning horizon $T = 19.29$.

orders, which prevents customer 3 the higher revenue to place its order. When the time slot assignment is unrestricted, this situation is averted. We observe this effect also in larger instances, even when customers have equal revenue $r = 1$ and probability $p = \frac{1}{2}$.

3 Expected Revenue Calculation

In this section, we present two algorithms for calculating the expected revenue of a given a priori route and time slot assignment. Calculating the expected revenue of a given solution efficiently is of critical importance to our solution methods. While sampling can be used to approximate the expected revenue, it turns out that exact calculation of the expected revenue, over all possible scenarios, can be done quite efficiently.

Obtaining the revenue of a single scenario ω, i.e., a sequence of arriving customers, can be done by checking whether an arriving customer can feasibly be inserted in the (partial) delivery route, and, if so, inserting it into the route. This requires $\mathcal{O}(k^2)$ operations for a scenario with k arriving customers. However, the number of scenarios $|\Omega|$ gets extremely large quickly, as the number of scenarios for an instance with n customer locations is $\mathcal{O}(n!)$, making brute-force enumeration computationally prohibitive. Fortunately, the observations presented earlier allow us to reduce the number of operations required substantially.

In the following, we present a naive enumeration algorithm and a customized dynamic programming algorithm for calculating the expected revenue of an a priori route. Both algorithms exploit the observations presented in Section 2.1.2.

3.1 Enumeration Algorithm

The expected revenue $\bar{r}$ of a given a priori route ρ and time slot assignment y can be calculated as follows:

$$\bar{r} = \sum_{\omega \in \Omega} \bar{r}_\omega = \sum_{S \subseteq \mathcal{V}_\mathrm{c}} \sum_{\omega \in \mathrm{Perm}\,(S)} \bar{r}_\omega = \sum_{S \subseteq \mathcal{V}_\mathrm{c}} \sum_{\omega \in \mathrm{Perm}\,(S)} p_\omega r_\omega, \tag{1}$$

with Perm (S) the set of all permutations of set of customers S, and $\bar{r}_\omega$ the probability-weighted revenue collected in scenario ω. Using our problem assumptions on the probability distribution of $\omega \in \Omega$, the probability p_ω of a single scenario ω, which is an ordered sequence of customer arrivals, can be determined as follows:

$$p_\omega = \prod_{i \in \omega} p_i \prod_{j \in \mathcal{V}_\mathrm{c} \setminus \omega} (1 - p_j) \frac{1}{|\omega|!}, \tag{2}$$

with $|\omega|$ the number of customers wanting to place an order in scenario ω. The first part of the expression on the right-hand side represents the probability that the customers in ω want to place an order and the second part of the expression on the right-hand side represents the probability that they do so in the sequence specified by ω. The revenue collected in scenario ω is given by

$$r_\omega = \sum_{i \in \omega} r_i z_i^\omega, \tag{3}$$

with z_i^ω for customer i in ω an indicator variable specifying whether or not customer i can place an order given the orders that have already been accepted from customers arriving before i in the sequence.

While the expected revenue can be calculated by summing the probability-weighted revenues $\bar{r}_\omega$, for every scenario $\omega \in \Omega$, the enumeration algorithm exploits the following observation to reduce the number of terms in the summation. For every subset $S \subseteq \mathcal{V}_\mathrm{c}$ of customers, if all can be feasibly inserted together in the a priori route with the time slot assignment, then all customer arrival sequences which are a permutation of subset S will result in the same delivery route and thus give the same revenue. In particular, the contribution $\bar{r}_S$ to the expected revenue of all permutations of subset S, which is the sum $\bar{r}_S = \sum_{\omega \in \mathrm{Perm}\,(S)} \bar{r}_\omega$, in that case reduces to

$$\begin{aligned} \bar{r}_S &= \prod_{i \in S} p_i \prod_{j \in \mathcal{V}_\mathrm{c} \setminus S} (1 - p_j) \sum_{\omega \in \mathrm{Perm}\,(S)} \frac{1}{|S|!} \sum_{i \in S} r_i \\ &= \prod_{i \in S} p_i \prod_{j \in \mathcal{V}_\mathrm{c} \setminus S} (1 - p_j) \sum_{i \in S} r_i, \end{aligned} \tag{4}$$

where we use the fact that all permutations ω of S are equally likely.

The enumeration algorithm evaluates every subset $S \subseteq \mathcal{V}_\mathrm{c}$, and starts by checking only one particular permutation ω of S. If all customer locations in ω can be inserted together in the given a priori route and time slot assignment, then the running expected revenue, the partial sum of $\bar{r}$, is increased by $\bar{r}_S$ given by (4) and the algorithm continues evalutating another subset S. However, if one or more customers in ω cannot be feasibly inserted together in the given a priori route and time slot assignment, the enumeration algorithm needs to calculate the contribution $\bar{r}_\omega = p_\omega r_\omega$ to the running expected revenue of each permutation ω of the subset S separately using (2) and (3).

For n customer locations, there are 2^n possible unordered subsets of customers and $|\Omega| = \sum_{k=0}^{n} \frac{n!}{k!}$ possible ordered subsequences of customer arrivals (see Sequence A000522 in Sloane, 2010). All unordered subsequences are checked by the enumeration algorithm, and typically not all ordered subsequences. However, in worst-case all ordered subsequences need to be checked. The naive enumeration algorithm therefore has a worst-case complexity of $\mathcal{O}\Big(\sum_{k=0}^{n} \frac{n!}{k!} \cdot n^2\Big) = \mathcal{O}\big(e \cdot n! \cdot n^2\big) = \mathcal{O}\big((n+2)!\big)$. We notice that the run-time of this algorithm suffers from the many redundant calculations that are done. These redundant calculations can be avoided by using a dynamic programming label extension algorithm.

3.2 DP Label Extension Algorithm

The Dynamic Programming algorithm exploits the following observations to reduce the number of operations needed for the revenue calculation. For any two scenarios ω_1 and ω_2, sequences of arriving customers (willing to place an order), which have equal sets of arriving customers (but in a different order) and equal sets of placed customers, i.e., their resulting delivery routes are equal, will also have the same contribution to the expected revenue, i.e., $\bar{r}_{\omega_1} = \bar{r}_{\omega_2}$. Moreover, in this case, the scenario $\omega_1' = \omega_1 \circ \{i\}$, in which customer i arrives after the customers of scenario ω_1, will have the same placed customer set, resulting delivery route and contribution to the expected revenue as scenario $\omega_2' = \omega_2 \circ \{i\}$, for each customer $i \in \mathcal{V}_\mathrm{c} \setminus \omega_1 = \mathcal{V}_\mathrm{c} \setminus \omega_2$.

These observations lead to the following label definition. A label $L = \Big(\mathcal{V}^{\mathrm{arr}}, \mathcal{V}^{\mathrm{placed}}, m\Big)$ is characterized by: (1) a subset $\mathcal{V}^{\mathrm{arr}}(L) \subseteq \mathcal{V}_\mathrm{c}$ of arrived customers (wanting to place an order), (2) a subset $\mathcal{V}^{\mathrm{placed}}(L) \subseteq \mathcal{V}^{\mathrm{arr}}(L)$ of placed customers (arrived and able to place their order), and (3) a multiplier $m(L)$ denoting the total number of scenarios which share the same subsets of arrived and placed customers. Notice that the contribution $\bar{r}(L)$ of a single label L, which represents all scenarios that share the subsets of arrived and placed customers, to the expected revenue is:

$$\bar{r}(L) = \prod_{i \in \mathcal{V}^{\mathrm{arr}}(L)} p_i \prod_{j \in \mathcal{V}_\mathrm{c} \setminus \mathcal{V}^{\mathrm{arr}}(L)} (1 - p_j) \frac{m(L)}{|\mathcal{V}^{\mathrm{arr}}(L)|!} \sum_{i \in \mathcal{V}^{\mathrm{placed}}(L)} r_i. \tag{5}$$

The expected revenue $\bar{r}$ for the given a priori route and time slot assignment is the sum of all label contributions $\bar{r}(L)$:

$$\bar{r} = \sum_{k=0}^{n} \sum_{L \in \mathcal{L}_k} \bar{r}(L), \tag{6}$$

with $\mathcal{L}_k$ the set of all possible labels with exactly $|\mathcal{V}^{\mathrm{arr}}(L)| = k$ arriving customers.

It is convenient to also record the probability $p(L)$ of the scenarios and the sum of revenues $r(L)$ of the placed customers in the label:

$$p(L) \quad = \quad \prod_{i\in\mathcal{V}^{\text{arr}}(L)} p_i \prod_{j\in\mathcal{V}_{\text{c}}\setminus\mathcal{V}^{\text{arr}}(L)} (1-p_j), \tag{7}$$

$$r(L) \quad = \quad \sum_{i\in\mathcal{V}^{\text{placed}}(L)} r_i. \tag{8}$$

The contribution $\bar{r}(L)$ of label L to the expected revenue then becomes

$$\bar{r}(L) \quad = \quad p(L)\frac{m(L)}{|\mathcal{V}^{\text{arr}}(L)|!}r(L). \tag{9}$$

Algorithm 1 shows the Dynamic Programming algorithm for calculating the expected revenue $\bar{r}$ over all scenarios exactly for a given a priori route ρ and time slot assignment y by extending labels. Here, the label sets $\mathcal{L}_k$, for $k \in \{0,1,\ldots,n\}$, includes all labels L with $|\mathcal{V}^{\text{arr}}(L)| = k$ number of arrived customers. The contribution of each label L in $\mathcal{L}_k$ is added to the running expected revenue $\bar{r}$, and then each label L is extended by adding a customer i not yet arrived to that label. Then, it is checked if customer i can be inserted in the a priori route ρ with time slot assignment y given the already placed customers in label L. The set of placed customers and the sum of placed customers are updated accordingly. Then, it is checked if a label $\tilde{L}$ with the same subset of arrived customers and the same subset of placed customers already exists in the label set $\mathcal{L}_{k+1}$. If so, the multiplier of the current label is added to that of the existing label. If not, a new label is added to the label set $\mathcal{L}_{k+1}$.

Algorithm 1 Dynamic Programming Algorithm

```
Input: A priori route $\rho$ and time slot assignment $y$.
Output: Exact expected revenue $\bar{r}$.
 1: $\bar{r} \leftarrow 0$
 2: $\mathcal{V}^{\text{arr}} \leftarrow \varnothing$, $\mathcal{V}^{\text{placed}} \leftarrow \varnothing$, $p \leftarrow \prod_{i\in\mathcal{V}_{\text{c}}}(1-p_i)$, $m \leftarrow 1$, $r \leftarrow 0$
 3: $\mathcal{L}_0 \leftarrow \left\{\left(\mathcal{V}^{\text{arr}}, \mathcal{V}^{\text{placed}}, p, m, r\right)\right\}$
 4: for $k = 0,1,\ldots,n$ do
 5:     for each $L = \left(\mathcal{V}^{\text{arr}}, \mathcal{V}^{\text{placed}}, m, p, r\right) \in \mathcal{L}_k$ do
 6:         $\bar{r} \leftarrow \bar{r} + p \cdot \frac{m}{|\mathcal{V}^{\text{arr}}|!} \cdot r$
 7:         for each $i \in \mathcal{V}_{\text{c}} \setminus \mathcal{V}^{\text{arr}}$ do
 8:             $\tilde{\mathcal{V}}^{\text{arr}} \leftarrow \mathcal{V}^{\text{arr}} \cup \{i\}$
 9:             if IsFeasibleInsertion($i$, $L$, $\rho$, $y$) then
10:                 $\tilde{\mathcal{V}}^{\text{placed}} \leftarrow \mathcal{V}^{\text{placed}} \cup \{i\}$
11:                 $r' \leftarrow r + r_i$
12:             else
13:                 $\tilde{\mathcal{V}}^{\text{place}} \leftarrow \mathcal{V}^{\text{placed}}$
14:                 $r' \leftarrow r$
15:             if there is an $\tilde{L} \in \mathcal{L}_{k+1}$ with $\mathcal{V}^{\text{arr}}(\tilde{L}) = \tilde{\mathcal{V}}^{\text{arr}}$ and $\mathcal{V}^{\text{placed}}(\tilde{L}) = \tilde{\mathcal{V}}^{\text{placed}}$ then
16:                 $m(\tilde{L}) \leftarrow m(\tilde{L}) + m$
17:             else
18:                 $p' \leftarrow p \cdot \frac{p_i}{1-p_i}$
19:                 $\mathcal{L}_{k+1} \leftarrow \mathcal{L}_{k+1} \cup \left\{\left(\tilde{\mathcal{V}}^{\text{arr}}, \tilde{\mathcal{V}}^{\text{placed}}, m, p', r'\right)\right\}$
```

The following Theorem characterizes the number of operations required by the algorithm. We are able to give a tight lower bound and an upper bound on the number of operations. It is currently unknown to us if the upper bound is tight.

Theorem 5. *The number of operations required by the Dynamic Programming Algorithm for an a priori route ρ and time slot assignment y with $|\mathcal{V}_c| = n$ customer locations is at least $\mathcal{O}(2^n n^2)$, but not more than $\mathcal{O}(3^n n^2)$.*

Proof. Each label is uniquely characterized by its set of arrived customers and set of placed customers. This means that each label set $\mathcal{L}_k$ contains at most one label with a unique combination of both sets. Furthermore, the label set $\mathcal{L}_k$, for $k \in \{0, 1, \ldots, n\}$, contains only labels L with a number of k arrived customers: $|\mathcal{V}^{\text{arr}}(L)| = k$. There are at most $\binom{n}{k}$ possible different sets of arriving customers of size k. Also, for each unique set of arriving customers $\mathcal{V}^{\text{arr}}(L)$, there are at most 2^k possible set of placed customers $\mathcal{V}^{\text{placed}}(L)$, since $\mathcal{V}^{\text{placed}}(L) \subseteq \mathcal{V}^{\text{arr}}(L)$ holds for any label L. Therefore, the total number of labels in $\mathcal{L}_k$ for $k \in \{0, 1, \ldots, n\}$ does not exceed $\binom{n}{k}2^k$. We note that this number is not necessary tight for every k. Each label $L \in \mathcal{L}_k$ has $|\mathcal{V}_c \setminus \mathcal{V}^{\text{arr}}(L)| = n - k$ possible extensions, which are all checked in the algorithm. Checking if a single customer can be feasibly inserted at a given position in a delivery route (containing the placed customers) can be done in $\mathcal{O}(k)$ operations. Alternatively, earliest and latest start of service times of the placed customers in the delivery route of a label can be kept in memory, reducing the number of operations required by the feasibility check to $\mathcal{O}(1)$. However, if the insertion is feasible, these earliest and latest start of service times need to be updated, which still requires $\mathcal{O}(k)$ operations and thus only lowers the absolute number of operations. Then, the algorithm checks if a label $\tilde{L}$ already exists with the same set of arrived customers and the same set of placed customers. We assume the labels in the label sets $\mathcal{L}_k$ are stored in a data structure which allows search and insertion to be done in logarithmic time complexity, for instance a binary tree data structure. This allows a number of $|\mathcal{L}_{k+1}| \leq \binom{n}{k+1}2^{k+1}$ stored labels to be searched in $\mathcal{O}\Big(\log\big(\binom{n}{k+1}2^{k+1}\big)\Big) = \mathcal{O}\Big(2\log 2^{k+1}\Big) = \mathcal{O}(k)$ operations, and a new label can be added in the same number of operations complexity. Alternatively, two nested binary tree data structures, one for the possible sets $\mathcal{V}^{\text{arr}}$ and one for the possible sets $\mathcal{V}^{\text{placed}}$, can be used. This results in the same number of operations: $\mathcal{O}\Big(\log\big(\binom{n}{k+1}\big) + \log 2^{k+1}\Big) = \mathcal{O}(k)$. Each stage $k \in \{0, 1, \ldots, n-1\}$ of the algorithm therefore requires no more than $\mathcal{O}\Big(\binom{n}{k}2^k(n-k)k\Big)$ operations, and in total the algorithms requires no more than $\mathcal{O}\Big(\sum_{k=0}^{n}\binom{n}{k}2^k(n-k)k\Big) = \mathcal{O}(3^n n^2)$ operations. It is currently unknown by the authors if this number of operations is tight. However, a tight lower bound can be given by considering the following special case. Suppose that all arriving customers can be feasibly placed together in the a priori route, i.e., $\mathcal{V}^{\text{arr}}(L) = \mathcal{V}^{\text{placed}}(L)$ for any label L. In that case the number of possible labels $|\mathcal{L}_k| \leq \binom{n}{k}$ for $k \in \{0, 1, \ldots, n\}$, and therefore the number of operations needed by the algorithm is then at most $\mathcal{O}(\sum_{k=0}^{n}\binom{n}{k}(n-k)k) = \mathcal{O}(2^n n^2)$. □

Theorem 5 shows that the worst case number of operations needed by the Dynamic Programming algorithm, $\mathcal{O}(3^n n^2)$, is much lower than the worst case number of operations needed by the Enumeration algorithm, $\mathcal{O}(n! \cdot n^2)$. The Dynamic Programming algorithm does need more memory then the enumeration algorithm: at each stage $k \in \{0, 1, \ldots, n-1\}$ both label sets $\mathcal{L}_k$ and $\mathcal{L}_{k+1}$ need to be kept in memory, while the enumeration algorithm requires almost no memory at all. Computational experiments show that the reduction in the number of operations allows us to calculate the expected revenue of a priori routes and time slot assignments with $n = 12$ customer locations, over $|\Omega| \approx 1.3 \cdot 10^9$ scenarios, exactly in under 10 seconds of computation time, while the enumeration algorithm takes over 60 minutes.

4 Solution Method

We propose an Sample Average Approximation (SAA) Monte-Carlo approach to solve the Single-Vehicle problem of Section 2.1. This method uses sampling of scenarios to solve large stochastic programs. In the following, we present a two-stage stochastic program which solves the single-vehicle problem of Section 2.1 exactly, and then give an overview of the SAA approach.

4.1 Stochastic Programming Formulation

The single-vehicle variant can be formulated as a two-stage mixed-integer stochastic program. The a priori route ρ and time slot assignment y are the first stage decisions, and the evaluation of the revenue of placed customer orders is the second stage "decision". In the second stage, there are no recourse options, so it is purely an evaluation stage. The objective of the stochastic program is the revenue obtained by the placed customer orders averaged over all possible scenarios. We now formulate the first stage, then the second stage and afterwards provide the objective of the stochastic program.

We need the following decision variables for the first stage. Let binary variables x_{ij} for arc $(i, j) \in \mathcal{A}$ be one if the arc is used in the a priori route, and zero otherwise. Let binary variables y_{is} for customer location $i \in \mathcal{V}_c$ and time slot $s \in \mathcal{T}_i$ be one if this time slot is assigned to this location, and zero otherwise. These are the main decision variables of the first stage. For subtour elimination constraints we introduce single commodity flow variables f_{ij}, which represent the position of location $i \in \mathcal{V}$ in the a priori route if arc $(i, j) \in \mathcal{A}$ is used in the a priori route, and are zero otherwise. Note that variables f_{ij} are fixed when binary variables x_{ij} are fixed. The first stage of the stochastic program can be formulated as follows:

$$\sum_{j \in \mathcal{V} \setminus \{o,i\}} x_{ij} = 1 \qquad \forall i \in \mathcal{V}_c, \tag{10}$$

$$\sum_{i \in \mathcal{V} \setminus \{j,d\}} x_{ij} = 1 \qquad \forall j \in \mathcal{V}_c, \tag{11}$$

$$\sum_{j \in \mathcal{V}_c} x_{oj} = 1, \quad \sum_{i \in \mathcal{V}_c} x_{id} = 1, \tag{12}$$

$$\sum_{j \in \mathcal{V} \setminus \{o\}} f_{ij} - \sum_{j \in \mathcal{V} \setminus \{d\}} f_{ji} = 1 \qquad \forall i \in \mathcal{V}_c, \tag{13}$$

$$f_{oj} = 0, \qquad \forall j \in \mathcal{V}_c, \tag{14}$$

$$x_{ij} \leq f_{ij} \leq n x_{ij} \qquad \forall (i, j) \in \mathcal{A},\ i \neq o, \tag{15}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{A}, \tag{16}$$

$$\sum_{s \in \mathcal{T}_i} y_{is} = 1 \qquad \forall i \in \mathcal{V}_c, \tag{17}$$

$$y_{is} \in \{0, 1\} \qquad \forall s \in \mathcal{T}_i,\ i \in \mathcal{V}_c. \tag{18}$$

Essentially, the first stage consists of a TSP single-commodity flow formulation for the a priori route and the time slot assignment part. Constraints (10)–(12) ensure that the a priori route starts and ends at the fulfillment center and that every customer location is visited. Subtours in the a priori route are eliminated by Constraints (13)–(15). These are known as single-commodity flow constraints. Constraints (17) ensure that every customer location i is assigned exactly one time slot from the set of possible time slots $\mathcal{T}_i$.

The second stage of the stochastic program ensures that the expected revenue of an a priori route and time slot assignment is obtained. We introduce some additional notation. Let variable u_i denote the position of customer location $i \in \mathcal{V}_\mathrm{c}$ in the a priori route. Let the second stage binary decision variable z_i^ω for customer location $i \in \omega$ arriving in scenario ω denote if this customer can place an order in scenario ω using the first stage a priori route and time slot assignment solution. Furthermore, let the set of vertices $\mathcal{V}_\mathrm{c}^{\omega h} \subseteq \mathcal{V}_\mathrm{c}$ contain the first h customer locations that arrived in scenario ω, for $h \in \{1, \dots, |\omega|\}$. Also, let $\mathcal{V}^{\omega h} = \mathcal{V}_\mathrm{c}^{\omega h} \cup \{o, d\}$. Similarly, let the arc set $\mathcal{A}_\mathrm{c}^{\omega h}$ contain all arcs in $\mathcal{A}$ that are between vertices in $\mathcal{V}_\mathrm{c}^{\omega h}$ and the arc set $\mathcal{A}^{\omega h}$ contain all arcs in $\mathcal{A}$ that are between vertices in $\mathcal{V}^{\omega h}$. In the second stage, we essentially have for each scenario ω and for each number of arrived customers $h \in \{1, \dots, |\omega|\}$ in that scenario a partial graph which is used to find the current delivery (execution) route containing the currently placed customer locations. Binary decision variables $x_{ij}^{\omega h}$ are one if arc (i, j) is used in the delivery route of placed customers in scenario ω with the first h customers arrived. Continuous decision variables $t_i^{\omega h}$ denote the time at which service starts at location $i \in \mathcal{V}^{\omega h}$, or, in case of fulfillment center start vertex o and in case of fulfillment center end vertex d, the time of departure or arrival at the fulfillment center, respectively. In case a customer cannot place an order, the corresponding start of service time decision variable is not used.

The second stage of the stochastic program can be formulated as follows:

$$u_i = \sum_{j \in \mathcal{V} \setminus \{o,i\}} f_{ij} \qquad \forall i \in \mathcal{V}_\mathrm{c}, \tag{19}$$

$$\sum_{j \in \mathcal{V}^{\omega h} \setminus \{o,i\}} x_{ij}^{\omega h} = z_i^\omega \qquad \forall i \in \mathcal{V}_\mathrm{c}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{20}$$

$$\sum_{i \in \mathcal{V}^{\omega h} \setminus \{j,d\}} x_{ij}^{\omega h} = z_j^\omega \qquad \forall i \in \mathcal{V}_\mathrm{c}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{21}$$

$$\sum_{j \in \mathcal{V}^{\omega h} \setminus \{o\}} x_{oj}^{\omega h} = 1, \quad \sum_{i \in \mathcal{V}^{\omega h} \setminus \{d\}} x_{id}^{\omega h} = 1 \qquad \forall h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{22}$$

$$u_i + 1 \le u_j + n\left(1 - x_{ij}^{\omega h}\right) \qquad \forall (i,j) \in \mathcal{A}_\mathrm{c}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{23}$$

$$t_i^{\omega h} + t_{ij} \le t_j^{\omega h} + T\left(1 - x_{ij}^{\omega h}\right) \qquad \forall (i,j) \in \mathcal{A}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{24}$$

$$\sum_{s \in \mathcal{T}_i} a_s y_{is} \le t_i^{\omega h} \le \sum_{s \in \mathcal{T}_i} b_s y_{is} \qquad \forall i \in \mathcal{V}_\mathrm{c}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{25}$$

$$u_i \ge 1 \qquad \forall i \in \mathcal{V}_c, \tag{26}$$

$$z_i^\omega \in \{0, 1\} \qquad \forall i \in \omega,\ \omega \in \Omega, \tag{27}$$

$$x_{ij}^{\omega h} \in \{0, 1\} \qquad \forall (i,j) \in \mathcal{A}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega, \tag{28}$$

$$0 \le t_i^{\omega h} \le T \qquad \forall i \in \mathcal{V}^{\omega h},\ h \in \{1, \dots, |\omega|\},\ \omega \in \Omega. \tag{29}$$

The position variables u_i are set by constraints (19) using the first stage commodity flow variables f_{ij}. Constraints (20)–(22) ensure that the delivery route in scenario ω after the first h customer arrivals contains only locations of the placed customer, and start and end locations of the fulfillment center. Notice that the z_j^ω variables couple the different h stages: If a customer j arriving as hth customer in a scenario ω may place the order ($z_j^\omega = 1$), then the customer location must be included in the current delivery route, stage h, as well as all subsequent delivery routes of stages, $h+1,\dots,|\omega|$. Otherwise, if the customer j may not place the order ($z_j^\omega = 0$), then this location should not be visited in the current delivery route and also all subsequent delivery routes, i.e., stages $h, h+1,\dots,|\omega|$. Constraints (23) make sure that the placed customer locations in the delivery route follow their position in the a priori route. Furthermore, these constraints ensure that the delivery routes do not contain subtours. Start of service times along the (partial) delivery routes are set by constraints (24), and constraints (25) ensures that the start of service times of the placed customer locations are feasible with respect to their assigned time slot. Both sets of constraints also ensure that the delivery route respects the fulfillment center time window $[0,T]$ (planning horizon). As a result, arriving customers (willing to place their order) cannot place their order if including them in the current delivery route will violate the time slots- or the fulfillment center time window restrictions.

The objective of the stochastic program is given by

$$\max \sum_{\omega\in\Omega} p_\omega \sum_{i\in\omega} r_i z_i^\omega. \tag{30}$$

This is the expected revenue, which is obtained by summation of the revenues obtained in each scenario $\omega \in \Omega$ weighted by the probability of each scenario p_ω. These probabilities are given by (2).

The two stages, i.e., constraints (10)–(29), together with objective (30) form a stochastic program which can be solved as a MIP. Although this formulation can be solved for small instances, it has a "flaw": the MIP has full information of the complete customer sequence ω and can therefore *anticipate* not yet arrived customers. That is, even when a customer $i = \omega(h)$ arrives as the hth customer in ω and can be feasibly inserted in the current delivery route, the MIP may decide to prohibit the customer from placing an order, i.e., set $z_i^\omega = 0$. This can be beneficial as it may allow placement of high revenue customers arriving later in scenario ω. However, in our setting this look-ahead is not possible: If an arriving customer can feasibly be served, the customer is allowed to place an order. Forcing *non-anticipation* can be incorporated in the model, but requires the introduction of (many) variables and constraints. Essentially, earliest and latest arrival time decision variables, $e_i^{\omega h}$ and $l_i^{\omega h}$, respectively, allow determination of whether the insertion of an arriving customer $i' = \omega(h)$ will violate $e_{i'}^{\omega h} \le l_{i'}^{\omega h}$ given earliest and latest arrival times of the other locations in the current delivery route. This customer must place an order if the insertion is feasible, and a customer must be prohibited from placing an order if the insertion is not feasible. The non-anticipatory constraints are 'big-M' type constraints, as they are linearizations of constraints with $\max\{\cdot,\cdot\}$ and $\min\{\cdot,\cdot\}$ operations. The non-anticipatory constraints can be found in Appendix A. The stochastic program for our problem, i.e., with non-anticipation constraints, is therefore given by constraints (10)–(29) and constraints (37)–(60) together with the objective (30).

4.2 Sample Average Approximation

When the number of customer locations n gets large, the above stochastic programming formulation is intractable as the number of scenarios becomes prohibitive (it grows exponentially – $\mathcal{O}(n!)$). A Sample

Average Approximation (SAA) approach alleviates this issue by repeatedly solving the stochastic program with only a sampled subset of scenarios. We refer the interested reader to Kleywegt et al. (2002), who investigate the mathematical details of the method. For convenience, but also since our evaluation of the true solution objective differs slightly from the method described by Kleywegt et al. (2002), we summarize here the most important results.

Let Ω_N be a single sample consisting of N randomly sampled scenarios. Then the *sample problem* for our stochastic programming formulation becomes

$$R(\Omega_N) = \max \frac{1}{N} \sum_{\omega \in \Omega_N} r_i z_i^{\omega},$$

subject to constraints (10)–(29) and (37)–(60), in which the set of all scenarios Ω is replaced by the sample Ω_N.

In the SAA approach, we generate M samples Ω_N^1, Ω_N^2, ..., Ω_N^M, each containing N randomly sampled scenarios. For each sample Ω_N^m with $m = 1, \dots, M$, the sample problem is solved to obtain an optimal solution (ρ^m, y^m) with objective $R(\Omega_N^m)$. Among these M solutions (ρ^m, y^m), we pick the best solution (ρ^*, y^*) using exact evaluation of the expected revenue:

$$(\rho^*, y^*) \in \underset{m=1,\dots,M}{\arg\max} \left\{ r(\rho^m, y^m) \right\},$$

with $r(\rho^m, y^m)$ the exact expected revenue of solution (ρ^m, y^m). This can be calculated efficiently by our customized Dynamic Programming algorithm (see Section 3.2). We obtain an estimate of the optimality gap by using the average objective $\bar{R}$ over the M sample problems given by

$$\bar{R} = \frac{1}{M} \sum_{m=1}^{M} R(\Omega_N^m).$$

An estimation of the optimality gap is given by

$$\Delta = \bar{R} - r(\rho^*, y^*). \tag{31}$$

The variance $\sigma^2_{\bar{R}-r(\rho^*,y^*)}$ of this optimality gap estimator is estimated by

$$\sigma^2_{\bar{R}-r(\rho^*,y^*)} = \sigma^2_{\bar{R}} = \frac{1}{(M-1)M} \sum_{m=1}^{M} (R(\Omega_N^m) - \bar{R})^2.$$

Notice that this does not include any variance associated with the expected revenue of the solution, since we can determine this expected revenue exactly, while typically in an SAA approach the expected revenue is obtained by estimation. The variance of the optimality gap estimator can be used to construct a one-sided $(1-\alpha)$ confidence interval for the estimated optimality gap:

$$\bar{R} - r(\rho^*, y^*) + \Phi^{-1}(1-\alpha) \frac{\sigma_{\bar{R}}}{\sqrt{M}}, \tag{32}$$

with Φ^{-1} the inverse cumulative standard normal distribution function. This one-sided confidence interval is useful in assessing the quality of the current best solution (ρ^*, y^*) and providing statistical bounds on the objective value of the optimal solution.

5 Heuristic Methods

In this section, we present four heuristic methods for solving the strategic time slot management problem for a single vehicle. The first three simplify the stochastic programming formulation to make it more tractable, the last one is a simple pragmatic heuristic that ignores the uncertainty. The three simplifications of the stochastic programming formulation can be solved using the SAA approached presented in Section 4.2.

5.1 SAA with Fixed A Priori Route

In the first simplification, an a priori tour is generated independently and the SAA approach is used to assign a time slot to each customer location. Although we have shown that an *optimal* TSP tour, one that minimizes travel times, is not always the best a priori tour (Observation 3), it is likely to be a good a priori tour. Furthermore, obtaining an optimal TSP tour is not difficult for the instance sizes we are interested in ($n \leq 12$). Note that when the travel times are symmetric, the optimal TSP tour has to be made directed by choosing an orientation.

Once an a priori tour has been generated, the time slot assignment can be obtained using the stochastic program of Section 4.1 and fixing the decision variables related to the a priori tour. That is, all decision variables related to the a priori tour, x_{ij}, f_{ij}, u_i, are fixed, the first stage constraints reduce to only (17) and (18), and the second stage reduces to (20)–(29) and (37)–(60). The SAA approach presented in Section 4.2 can be used to solve this stochastic program, which is likely to be more efficient than solving the full stochastic program of Section 4.1.

It seems natural to use an *optimal* TSP tour, one that minimize the travel time, as a priori tour, but this not required.

5.2 SAA with Ascending Time Slots

In the second simplification, we enforce that the time slots are ascending along the a priori route, i.e., if the a priori route is $(o, 1, 2, \ldots, n, d)$, then $a_1 \leq a_2 \leq \ldots \leq a_n$. Although we have shown that ascending time slots along the a priori route is not always optimal (Observation 4), in most cases an optimal solution in which the time slots are ascending exists. Therefore, it is natural to consider a variant of our SAA approach in which the assigned time slots are restricted to being ascending along the a priori tour. We simply add the following constraints to the stochastic program:

$$\sum_{s\in\mathcal{T}} a_s y_{is} \leq \sum_{s\in\mathcal{T}} a_s y_{js} + T\,(1 - x_{ij}) \qquad \forall i, j \in \mathcal{V}_{\mathrm{c}}, i \neq j, \tag{33}$$

$$\sum_{s\in\mathcal{T}} b_s y_{is} \leq \sum_{s\in\mathcal{T}} b_s y_{js} + T\,(1 - x_{ij}) \qquad \forall i, j \in \mathcal{V}_{\mathrm{c}}, i \neq j, \tag{34}$$

Adding these constraints decreases the solution space, which is likely to result in a faster solution times.

5.3 SAA without Non-Anticipation

In the third simplification, we drop the non-anticipatory constraints (37)–(60). Most of these are 'big-M' constraints, which typically result in weak LP relaxation bounds and large search trees. Therefore, it might be computationally advantageous to use a variant of the stochastic program in which the non-anticipation constraints are omitted (i.e., only use (10)–(29)). Note that the bounds obtained in the SAA method now correspond to a relaxation of the problem and therefore do not provide insights in the optimal objective value of the original problem. However, the MIP might be easier to solve than the full stochastic program, and therefore might speed-up the SAA method.

5.4 A Linear Scaling Heuristic

The Linear Scaling Heuristic (LSH) is a two-phase heuristic (similar to SAA with Fixed A Priori Route) in which an a priori tour is generated first and then, given this a priori tour, time slot are assigned to each customer location using a simple linear scaling rule. The time slot assignment rule is based on the idea of linearly scaling the visit times of the customer locations in the a priori route to "fit" the planning horizon.

Before describing the time slot assignment rule, it is useful to introduce some additional notation. Let e_i and l_i for each customer location $i \in \mathcal{V}_c$ be the earliest and latest times that delivery can start at the customer location when directly visited after- and before the fulfillment center, respectively, i.e.,

$$e_i = a_o + t_{oi} = t_{oi},$$
$$l_i = b_d - t_{id} = T - t_{id},$$

for each $i \in \mathcal{V}_c$. Note that these quantities are independent of the a priori tour. We assume that $e_i \leq l_i$ holds for each customer location $i \in \mathcal{V}_c$, i.e., it is possible to service each customer location. The time slot to be assigned to customer location i needs to have overlap with $[e_i, l_i]$, or else the customer location cannot be served in tour and will always be rejected. By definition, the set of possible time slots $\mathcal{T}_i$ of customer location i contains only time slots which have overlap with $[e_i, l_i]$. Let the a priori route be $\rho^{\text{TSP}} = (o, 1, 2, \ldots, n, d)$. Note that, typically, the a priori route is not feasible itself, i.e., the length of the a priori route is more than T. As we have seen in Observation 1, in case the a priori route is feasible, then it is optimal to assign each customer location i a time slot which contains t_i^{TSP}, the time the a priori route visits customer location i.

LSH linearly scales down times t_i^{TSP} to ensure that these times lie within the planning horizon $[0, T]$. The resulting scaled times $\tilde{t}_i$ are then used to assign the time slot $s_i^* \in \mathcal{T}_i$ to customer location i. More specifically, we obtain the scaled time $\tilde{t}_i$ for customer location $i \in \mathcal{V}_c$ as

$$\tilde{t}_i = t_i^{\text{TSP}} \cdot \frac{\min\{T, t_d^{\text{TSP}}\}}{t_d^{\text{TSP}}}.$$

Should the scaled time $\tilde{t}_i$ lie outside the interval $[e_i, l_i]$, then we adjust it to the boundary of the interval, i.e.,

$$\tilde{t}_i = \begin{cases} l_i & \text{if } \tilde{t}_i > l_i, \\ \tilde{t}_i & \text{if } \tilde{t}_i \in [e_i, l_i], \\ e_i & \text{if } \tilde{t}_i < e_i. \end{cases}$$

To select a time slot for customer i using the scaled time $\tilde{t}_i$, we calculate a score c_{is} for each time slot $s \in \mathcal{T}_i$, and pick the time slot s^* with the highest score: $s^* = \arg\max_{s\in\mathcal{T}_i} c_{is}$. Should there be a tie between multiple time slots, then the time slot starting the earliest is chosen. We consider the following score function:

$$c_{is} = (\tilde{t}_i - a_s) \cdot (b_s - \tilde{t}_i).$$

This score function can be calculated quickly. It has the property that positives scores are given for scaled times inside the time slot and have a peak in the center of the time slot. Remember that we assume that the set of possible time slots $\mathcal{T}$ has time slots with equal width. Therefore, should the time slots in $\mathcal{T}$ be non-overlapping, then using these score functions simply results in selecting the time slot which has overlap with the scaled time. However, when the time slots in $\mathcal{T}$ overlap, then using these score functions results in selecting the time slot with its center closest to the scaled time.

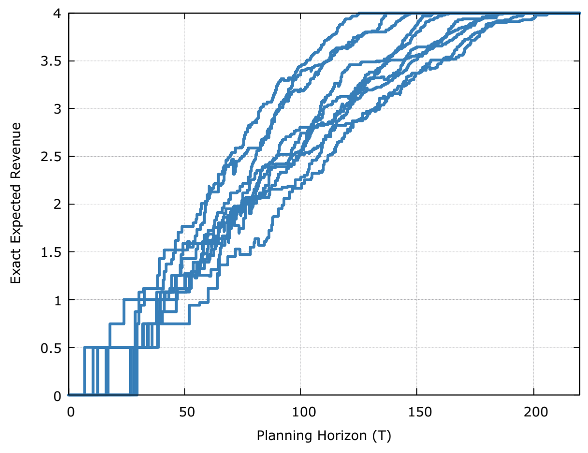
6 Computational Experiments

In this section, we present the results of numerical experiments in which we use the presented methods on randomly generated instances. The methods have been implemented in Python version 2.7.11, and the MIPs are solved using Gurobi version 8.0.1 with the default settings. The runs are executed as a single thread on an Intel® Xeon® E5-2650 v2 with 2.6 GHz (Turbo Boost up to 3.6 GHz) and 32 GB of RAM running Debian Linux version 9. All CPU times reported have been obtained using the wall-clock timer `timeit.default_timer()`. Only a single Python thread was active at any time on the CPU, while the MIP solver Gurobi had access to all available CPU cores.

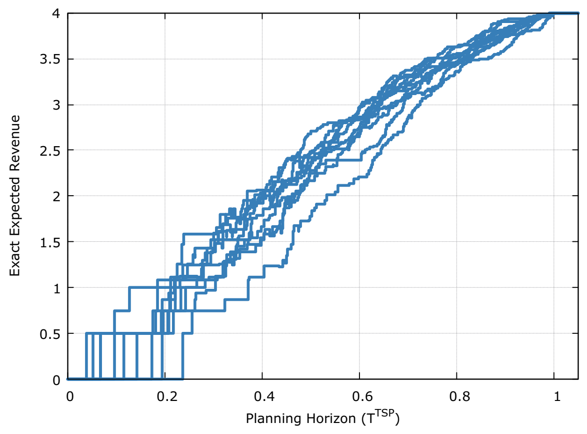
6.1 Instance Generation and Parameters

We generated sets of 10 instances for $n = 4$, 8, and 12 customer locations. These locations have integer coordinates which are randomly uniform in a $[0, 60] \times [0, 60]$ square. The fulfillment center is located in the center of the region (coordinates $(30, 30)$). The travel times between two locations is taken to be the Euclidean distance rounded up to two decimals and the service time at a location is taken to be zero. The customers have equal order placement probability $p = 0.5$ and result in equal revenue $r = 1$.

For each instance, it is also necessary to specify a horizon T and set of time slots $\mathcal{T}$. To account for the fact that not all customers will place an order, the company may design an a priori route with a duration that exceeds the operating horizon T. Depending on the company's risk tolerance (and service level targets), the company may choose an a priori route with a duration that results in an expected delivery route length of T or less (where the expected delivery route length is a function of the order placement probabilities of the customers). Indeed, by Observation 1, a horizon T equal to the minimum duration tour visiting all customer locations T^{TSP} implies that all customers can place an order (regardless of the sequence in which they place an order) and the company has no risk (while the actual delivery routes may vary in duration). Therefore, to reflect these choices, we model the company's risk tolerance, e.g., low, medium, and, high, by setting the horizon T as a function of T^{TSP}, specifically $T = 0.9T^{\text{TSP}}$, $T = 0.75T^{\text{TSP}}$, and $T = 0.6T^{\text{TSP}}$. (That is, to simplify instance generation, rather than designing instances for which the a priori route has duration of $\frac{1}{0.9}T$, $\frac{1}{0.75}T$, or $\frac{1}{0.6}T$, we do the reverse.) Thus, we generate three "regimes" of instances, characterized by the associated risk tolerance. Note that within a regime (set of instances), the duration T^{TSP} can vary substantially. Instances with locations that are relatively close together typically have a shorter a priori tour than instances with locations that are relatively far apart. The benefit of setting the horizon as a fraction of the minimum duration tour rather than as an absolute value is highlighted in Figure 5. The expected revenue is more consistent when the horizon is set



(a) Absolute.



(b) Factor of T^{TSP}.

Figure 5: Exact expected revenues for the c60 instances with 8 customers using the minimum duration tour as a priori route and no time slots, $\mathcal{T} = \{[0, T]\}$, plotted using (a) an absolute horizon and (b) plotted using a horizon relative to the duration of the optimal tour (T^{TSP}).

as a fraction of the minimum duration tour rather than as an absolute value.

We consider two time slot sets $\mathcal{T}$: non-overlapping and overlapping. Both sets have time slots with a fixed width w. The non-overlapping time slot set is given by

$$\mathcal{T} = \{[0, w], [w, 2w], [2w, 3w], \ldots\}, \tag{35}$$

and the overlapping time slot set is given by

$$\mathcal{T} = \left\{[0, w], \left[\frac{1}{2}w, \frac{3}{2}w\right], [w, 2w], \left[\frac{3}{2}w, \frac{5}{2}w\right], \ldots\right\}. \tag{36}$$

The latter set has two overlapping time slots for each time $t \in [0, T]$. Similar to the horizon, we choose the time slot width as fraction of the duration of the a priori route, i.e., $w = 0.25T^{\text{TSP}}$ and $0.125T^{\text{TSP}}$. As a benchmark, we also consider the case of "no slots", in which essentially each customer locations gets a time slot equal to the full horizon, i.e., a possible time slot set of $\mathcal{T} = \{[0, T]\}$.

The instances generated are recorded in VRP-REP XML format (Mendoza et al. (2014), see `http://vrp-rep.org`). We also include an optimal directed TSP tour and its duration. This directed TSP tour is used by the methods that require a fixed a priori route. The TSP tour is obtained by solving the formulation (10)–(16) with the objective of minimizing tour duration. For convenience, the horizons and time slot sets are also included in each instance file.

All SAA methods are run with sample sizes varying from $N = 2$ up to $N = 64$ (depending on instance size). Each SAA run uses a fixed sample size and $M = 20$ repetitions. After solving each MIP, the exact expected revenue is calculated using the DP algorithm presented in Section 3.2. During the run, the best solution found so far is kept in memory and used as starting solution for the MIP solver.

Not surprisingly, the full SAA method described in Section 4 is very demanding computationally. Therefore, we decided to run the SAA with Fixed A Priori Route method one time ($M = 1$) using the same sample and use the solution obtained as starting solution for the first MIP of the full SAA method run. This computation time is included when reporting solution time for the full SAA method. In our experiments, we solve all MIPs to optimality, i.e., no time limit was set. (Imposing a time limit increased the optimality gap estimates and decreased the solution quality significantly.)

6.2 Expected Revenue Calculation

We use exact calculation of the expected revenue in all our presented methods. While the Linear Scaling Heuristic of Section 5.4 requires only one evaluation at the end to obtain the expected revenue, the SAA methods require M evaluations. Thus, for the SAA methods, it is critical that the expected revenue calculation are done efficiently, in a reasonable amount of time.

In Table 1, we compare the computing times of the enumeration algorithm (ENUM) presented in Section 3.1 with the customized dynamic programming algorithm (DP) presented in Section 3.2 on instances of different sizes, where the Linear Scaling Heuristic (LSH) was used to obtain the solutions. Both algorithms were given a time limit of 1 hour for evaluation of the expected revenue of each instance. The time slot width w is set to $0.25T^{\text{TSP}}$ and the horizon T is set to $0.90T^{\text{TSP}}$, $0.75T^{\text{TSP}}$ and $0.60T^{\text{TSP}}$. We also report the total number of scenarios $|\Omega|$ over which the expected revenue is calculated. The computing times reported are averages over the 10 instances in a set and given in seconds. For convenience, we also report the speed-up of the DP algorithm over the ENUM algorithm.

Table 1: Comparison of the computing times required to evaluate the exact expected revenue using the enumeration algorithm and the dynamic programming algorithm.

					CPU time (s)		Speed up
	n	$\|\Omega\|$	w	T	ENUM	DP	DP
c60	4	65	0.25	0.90	0.0012	0.0031	0.38
			0.25	0.75	0.0011	0.0034	0.33
			0.25	0.60	0.0015	0.0036	0.42
c60	8	109601	0.25	0.90	4.42	0.12	37.87
			0.25	0.75	4.16	0.16	25.35
			0.25	0.60	3.85	0.17	22.31
c60	12	$1.3 \cdot 10^9$	0.25	0.90	> 3600	2.58	> 1398
			0.25	0.75	> 3600	4.80	> 750
			0.25	0.60	> 3600	6.07	> 592

While the ENUM algorithm is slightly faster on small instances with 4 customer locations, the DP algorithm outperforms the ENUM algorithm on mid and large size instances with 8 and 12 customer locations. Moreover, we see that the ENUM algorithm reaches the time limit of 1 hour for the evaluation of the expected cost of solutions for instance with 12 customer instances, while the DP algorithm takes less than 10 seconds for these instances. We observe too that the computing time depends on the horizon T. Whereas the ENUM algorithm requires less computing time for instances with a shorter horizon T, the DP algorithm requires more computing time for instances with a shorter horizon T. This is probably due to the fact that the required number operations, which we investigated in Section 3.2, tends towards the lower bound $\mathcal{O}(2^n n^2)$ in case of a long horizon, while it tends more to the upper bound $\mathcal{O}(3^n n^2)$ in case of a short horizon.

6.3 Non-overlapping Time Slots

LSH (Section 5.4), SAA with Fixed A Priori Route (Section 5.1) and Full SAA (Section 4) are tested on the instance sets with non-overlapping time slot sets (35). The expected revenue of the best found solution and the computation time in seconds, both averaged over each set of 10 instances, are presented in Table 2. We also report the expected revenue when the a priori tour is chosen to be the minimum duration tour and each location can be visited at any time during the horizon ("No Slots"). This expected revenue serves as an upper bound on the expected revenue, although it must be noted that it is not an exact upper bound. The time slot width w is set to $0.25T^{\text{TSP}}$ and $0.125T^{\text{TSP}}$, and the horizon T is set to $0.90T^{\text{TSP}}$, $0.75T^{\text{TSP}}$ and $0.60T^{\text{TSP}}$. For $n = 4$ customers, sample size $N = 64$ is used for both SAA with Fixed A Priori Route and Full SAA. For these small instances, however, the complete stochastic programs of Section 4.1 containing all scenarios ($|\Omega| = 65$) can be solved for both variants, i.e., without sampling. While both SAA methods and the complete stochastic programs produce solutions with the same expected revenue, the computation times differ since the SAA methods rely on solving $M = 20$ MIPs, while the the complete stochastic programs are only solved once. In Table 2, the marked computation times (*) where obtained by solving the complete stochastic program once, rather than using the SAA method. For $n = 8$ customers, sample size $N = 64$ is used for SAA with Fixed A Priori Route and $N = 16$ for Full SAA. For $n = 12$ customers, sample size $N = 32$ is used for SAA with Fixed A Priori Route. Unfortunately, Full SAA could not solve these instances in a reasonable amount of time.

As expected, we see that the expected revenue depends on the horizon T, and that it increases for longer horizons. This is also illustrated in Figure 6 for the $n = 8$ customer instances. Not surprisingly, the best found solutions are obtained by Full SAA. Interestingly, the expected revenues obtained by Full SAA approach the No Slots upper bound. While SAA with Fixed A Priori Route does not always find the best solutions, the revenues obtained are on average only 3% worse and the computation times are much smaller. This indicates that the optimal TSP tour is good a priori tour. LSH obtains solutions that are on average 7% lower than the best found solutions, but the method requires relatively little computation time. Notice that the computation time required by LSH essentially consists of the single expected revenue calculation. LSH performs especially well in case of a long horizon $T = 0.90T^{\text{TSP}}$. When decreasing the time slot width w, from $0.25T^{\text{TSP}}$ to $0.125T^{\text{TSP}}$, the expected revenue of the best found solutions (using Full SAA) decreases by only 1 – 2%. Also, the solutions obtained by SAA with Fixed A Priori Route have 1 – 2% lower expected revenue when lowering the time slot width, while the solutions obtained by LSH are

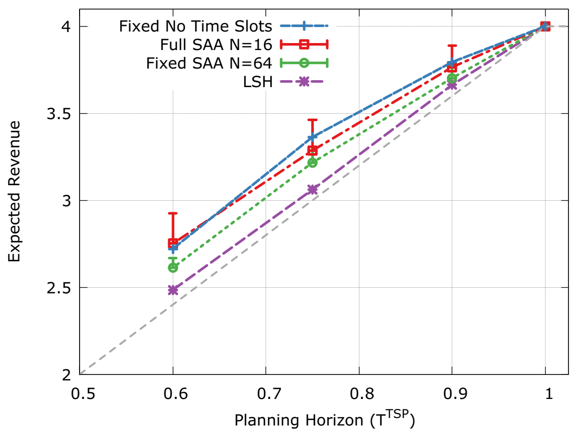
Table 2: Results for the instances with non-overlapping time slots.

				Exact Exp. Rev.				CPU time (s)		
	n	w	T	LSH	SAA Fixed	SAA Full	No Slots	LSH	SAA Fixed	SAA Full
c60	4	0.25	0.90	1.73	1.75	1.79	1.79	0.00	1.49*	25.57*
		0.25	0.75	1.51	1.54	1.56	1.59	0.00	1.43*	28.36*
		0.25	0.60	1.26	1.28	1.31	1.31	0.00	0.82*	6.46*
c60	4	0.125	0.90	1.65	1.70	1.73	1.79	0.00	4.84*	70.78*
		0.125	0.75	1.39	1.53	1.55	1.59	0.00	2.69*	31.58*
		0.125	0.60	1.14	1.27	1.30	1.31	0.00	1.02*	9.61*
c60	8	0.25	0.90	3.66	3.70	3.77	3.79	0.12	166.66	3018.92
		0.25	0.75	3.06	3.22	3.29	3.36	0.16	268.51	6248.68
		0.25	0.60	2.49	2.61	2.75	2.72	0.17	59.94	1429.13
c60	8	0.125	0.90	3.60	3.63	3.72	3.79	0.12	724.26	4752.46
		0.125	0.75	2.90	3.18	3.27	3.36	0.16	735.78	5332.23
		0.125	0.60	2.28	2.59	2.72	2.72	0.17	246.61	1077.33
c60	12	0.25	0.90	5.56	5.69		5.81	2.58	303.39	
		0.25	0.75	4.79	4.96		5.17	4.80	649.04	
		0.25	0.60	3.90	4.05		4.23	6.07	337.70	
c60	12	0.125	0.90	5.44	5.63		5.81	2.82	1298.79	
		0.125	0.75	4.59	4.93		5.17	4.78	2204.55	
		0.125	0.60	3.67	4.02		4.23	5.73	994.01	

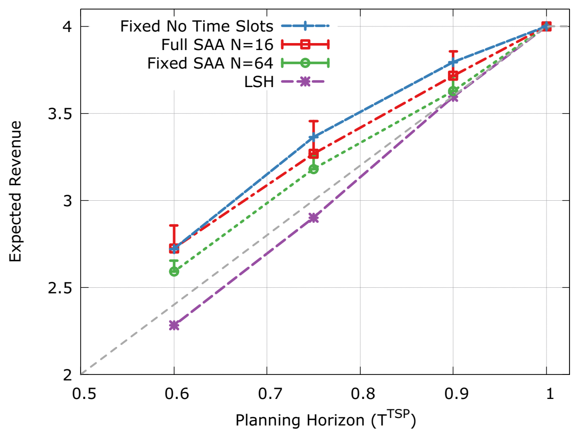
much worse. The time slot set $\mathcal{T}$ contains twice as many time slots compared to width $w = 0.25T^{\text{TSP}}$. It seems that the simple decision rule used by LSH is less effective when time slots are smaller. Moreover, the computation time needed by both SAA methods increases, also probably because the number of possible time slots increases.

The SAA methods rely on repeated solving MIPs containing only a sampled subset of N scenarios. This sample size N not only affects the quality of the obtained solution, but also the computation time needed, and the quality of the obtained estimate of the optimality gap. Table 3 compares the quality of the solution, the estimated optimality gap (Equation (31)) and the confidence interval of the optimality gap for sample sizes of $N = 2$ up to 64 obtained by Full SAA. The reported confidence interval is the rightmost term in Equation (32) with $\alpha = 0.05$. We observe from the table that the estimated optimality gaps and the confidence intervals decrease as the sample size increases. We observe too that for the highest sample sizes, the estimated gap and the confidence interval are both relatively small and of the same order of magnitude, making it likely that a (near-)optimal solution is found. The expected revenue does not increase much when increasing the sample size beyond a certain point, while computation times do increase noticeably. Similarly, Table 4 compares the quality of the solution, the estimated optimality gap (Equation (31)) and the confidence interval of the optimality gap for sample sizes of $N = 2$ up to 64 of SAA With Fixed A Priori Route. The estimated optimality gaps and confidence intervals decrease as the sample size increases. Compared to the Full SAA method, the computation times and the estimated optimality gaps are smaller, which indicates that this restricted problem is easier to solve. The interested reader is referred to Appendix B, which contains more detailed results of our experiments.

Figure 7 shows the relation between solution quality and computation time for the $n = 8$ customer instances with time slot width $w = 0.125T^{\text{TSP}}$, averaged over all instances in the set and all horizons $T \in \{0.60, 0.75, 0.90\}\, T^{\text{TSP}}$. The expected revenue is shown as a fraction of the expected revenue of the



(a) Time slot width $w = 0.25T^{\text{TSP}}$.



(b) Time slot width $w = 0.125T^{\text{TSP}}$.

Figure 6: Results for the instances with $n = 8$ customers and non-overlapping time slot sets. The error bars of the SAA methods indicate the average estimated optimality gap with the confidence interval. The grey dashed line highlights an (imaginary) linear relation between planning horizon and expected revenue.

best found solutions. The numbers above the points indicate the sample size N of the SAA methods. We see that the revenues associated with LSH solutions are on average around 90% of the revenues of the best found solutions, while SAA Fixed A Priori Route converges to around 97% of the revenues of the best found solutions. Sample sizes of $N = 16$ and higher contribute little to improve the solution quality compared to $N = 8$, while requiring more computation time. The best solutions are found by Full SAA. However, Full SAA performs worse than SAA Fixed A Priori Route for sample size $N = 2$. Samples with only two scenarios are likely too small to properly capture the stochastic information.

Table 3: Results of Full SAA with $M = 20$ samples. Each row shows averages over 10 instances with n customers and non-overlapping time slot sets.

				Exact Expected Revenue						Estimated Gap						Gap Confidence Interval ($\alpha = 0.05$)						CPU time (s)					
	n	w	T	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$
c60	4	0.25	0.90	1.71	1.78	1.79	1.79	1.79	1.79	0.15	0.04	-0.04	-0.02	-0.01	-0.00	0.22	0.15	0.11	0.07	0.06	0.04	3.59	1.57	3.13	7.91	26.14	86.85
		0.25	0.75	1.54	1.56	1.56	1.56	1.56	1.56	0.07	0.05	-0.00	0.03	0.03	0.01	0.19	0.14	0.09	0.07	0.04	0.03	7.90	1.22	2.53	6.12	18.46	58.67
		0.25	0.60	1.30	1.30	1.31	1.31	1.31	1.31	-0.01	-0.03	0.00	0.01	0.01	-0.00	0.16	0.10	0.07	0.06	0.04	0.03	0.65	0.86	1.52	3.30	7.90	19.17
c60	4	0.125	0.90	1.71	1.72	1.73	1.73	1.73	1.73	0.07	0.09	0.05	0.02	0.03	0.02	0.22	0.15	0.10	0.07	0.05	0.04	13.80	1.99	3.74	11.40	49.02	234.12
		0.125	0.75	1.52	1.54	1.55	1.55	1.55	1.55	0.13	0.08	0.05	-0.01	0.01	0.02	0.19	0.12	0.09	0.07	0.05	0.03	0.66	0.91	2.20	5.70	16.63	92.60
		0.125	0.60	1.26	1.30	1.30	1.30	1.30	1.30	0.06	-0.00	0.01	0.00	-0.00	0.00	0.15	0.12	0.08	0.05	0.04	0.02	0.60	0.71	1.30	2.72	6.43	22.26
c60	8	0.25	0.90	3.50	3.66	3.75	3.77			0.38	0.16	0.05	0.01			0.36	0.23	0.16	0.11			48.63	126.27	1065.00	3018.92		
		0.25	0.75	3.10	3.20	3.26	3.29			0.38	0.22	0.13	0.08			0.32	0.21	0.14	0.09			62.74	182.23	809.21	6248.68		
		0.25	0.60	2.54	2.68	2.74	2.75			0.34	0.21	0.10	0.08			0.26	0.17	0.14	0.09			45.07	74.39	243.42	1429.13		
c60	8	0.125	0.90	3.43	3.66	3.70	3.72			0.43	0.16	0.18	0.02			0.33	0.24	0.16	0.12			41.27	130.25	740.59	4752.46		
		0.125	0.75	2.97	3.19	3.25	3.27			0.52	0.28	0.18	0.09			0.28	0.20	0.15	0.09			29.01	127.16	584.37	5332.23		
		0.125	0.60	2.57	2.63	2.71	2.72			0.36	0.24	0.11	0.04			0.25	0.19	0.11	0.09			25.49	58.10	155.56	1077.33		

Table 4: Results of SAA with Fixed A Priori Route with $M = 20$ samples. Each row shows averages over 10 instances with n customers and non-overlapping time slot sets.

				Exact Expected Revenue						Estimated Gap						Gap Confidence Interval ($\alpha = 0.05$)						CPU time (s)					
	n	w	T	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$
c60	4	0.25	0.90	1.74	1.75	1.75	1.75	1.75	1.75	0.10	0.05	-0.02	-0.00	0.00	0.00	0.22	0.15	0.11	0.07	0.06	0.04	0.75	0.77	1.30	2.39	4.68	9.58
		0.25	0.75	1.54	1.54	1.54	1.54	1.54	1.54	0.07	0.05	-0.00	0.02	0.03	0.00	0.19	0.14	0.09	0.06	0.04	0.03	0.88	0.74	1.24	2.40	4.63	9.30
		0.25	0.60	1.28	1.28	1.28	1.28	1.28	1.28	0.01	-0.01	0.02	0.02	0.02	0.00	0.15	0.10	0.07	0.05	0.04	0.02	0.49	0.65	1.18	2.21	4.26	8.55
c60	4	0.125	0.90	1.70	1.70	1.70	1.70	1.70	1.70	0.07	0.08	0.04	0.02	0.03	0.01	0.22	0.14	0.10	0.07	0.05	0.04	1.74	0.89	1.60	2.97	6.99	21.96
		0.125	0.75	1.51	1.53	1.53	1.53	1.53	1.53	0.13	0.06	0.03	-0.03	0.00	0.00	0.19	0.12	0.09	0.07	0.05	0.03	0.51	0.81	1.44	2.65	5.20	17.47
		0.125	0.60	1.27	1.27	1.27	1.27	1.27	1.27	0.05	0.02	0.02	0.01	0.01	0.01	0.15	0.11	0.08	0.05	0.04	0.02	0.50	0.71	1.25	2.29	4.56	9.89
c60	8	0.25	0.90	3.57	3.70	3.70	3.70	3.70	3.70	0.27	0.07	0.03	0.01	0.01	0.02	0.35	0.23	0.16	0.11	0.08	0.05	4.24	5.45	8.58	16.74	40.77	166.66
		0.25	0.75	3.19	3.20	3.22	3.22	3.22	3.22	0.21	0.12	0.06	0.04	0.04	0.02	0.30	0.19	0.14	0.09	0.06	0.05	4.65	6.39	10.45	22.18	59.21	268.51
		0.25	0.60	2.57	2.60	2.61	2.61	2.61	2.61	0.19	0.15	0.08	0.06	0.03	0.01	0.25	0.16	0.13	0.08	0.06	0.04	4.95	6.11	8.89	14.75	27.39	59.94
c60	8	0.125	0.90	3.57	3.62	3.63	3.63	3.63	3.63	0.23	0.12	0.16	0.03	0.03	0.03	0.32	0.23	0.15	0.12	0.07	0.05	4.36	6.01	11.24	23.23	81.86	724.26
		0.125	0.75	3.06	3.18	3.18	3.18	3.18	3.18	0.33	0.17	0.12	0.06	0.04	0.03	0.27	0.19	0.15	0.09	0.07	0.05	4.81	6.72	11.78	26.16	77.38	735.78
		0.125	0.60	2.53	2.56	2.59	2.59	2.59	2.59	0.29	0.20	0.09	0.03	0.04	0.02	0.23	0.17	0.10	0.08	0.06	0.04	5.19	6.73	10.34	18.84	45.54	246.61
c60	12	0.25	0.90	5.50	5.68	5.69	5.69	5.69		0.24	0.12	0.02	0.02	0.03		0.45	0.30	0.21	0.16	0.09		63.80	57.72	65.10	106.05	303.39	
		0.25	0.75	4.80	4.94	4.96	4.95	4.96		0.46	0.19	0.15	0.08	0.04		0.37	0.26	0.17	0.13	0.09		86.74	88.76	112.23	202.94	649.04	
		0.25	0.60	3.96	4.02	4.04	4.05	4.05		0.51	0.27	0.17	0.12	0.08		0.32	0.22	0.15	0.11	0.08		103.15	108.98	120.03	161.55	337.70	
c60	12	0.125	0.90	5.32	5.62	5.63	5.63	5.63		0.56	0.28	0.11	0.07	0.02		0.42	0.30	0.18	0.13	0.10		67.38	61.96	81.77	217.91	1298.79	
		0.125	0.75	4.68	4.84	4.92	4.93	4.93		0.53	0.33	0.17	0.06	0.05		0.37	0.26	0.19	0.13	0.08		86.49	89.03	119.52	251.69	2204.55	
		0.125	0.60	3.87	3.98	4.00	4.02	4.02		0.57	0.33	0.22	0.03	0.07		0.31	0.22	0.16	0.11	0.09		100.80	104.58	127.63	204.72	994.01	

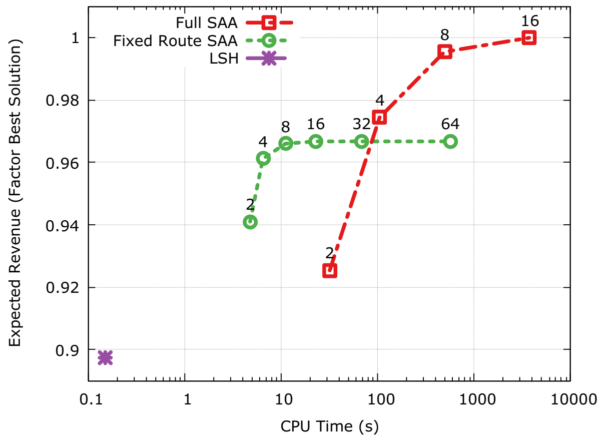


Figure 7: Solution quality versus computation time needed by the solution methods for solving the $n = 8$ customer instances with non-overlapping time slot of width $w = 0.125T^{\text{TSP}}$.

6.4 Overlapping Time Slots

To assess the benefit of having overlapping time slots, we also solve the instance sets with the time slot set (36) using LSH, SAA With Fixed A Priori Route and Full SAA. Table 5 shows the expected revenue and computation time in seconds for the methods, using the same sample sizes as in Table 2. We also report the expected revenue and computation time as a fraction of the corresponding values for the non-overlapping time slot set. The methods are tested on the instances with $n = 8$ and 12 customers, with time slot width $w = 0.25T^{\text{TSP}}$ and $0.125T^{\text{TSP}}$, and with horizons $T \in \{0.90, 0.75, 0.60\}\,T^{\text{TSP}}$. The column "Ov." indicates how many time slots in $\mathcal{T}$ are overlapping for any single point in time. We observe that the revenues increase by around $3 - 5$ % when using LSH, by around $2 - 3$ % when using SAA with Fixed A Priori Route and 1% – 2% when using Full SAA, while the best solutions are again obtained by Full SAA for the $n = 8$ customer instances. However, the computation times for the SAA methods increase dramatically: computation times increased 2 to 6 times when solving the instances with the overlapping time slot set compared to solving the instances with the non-overlapping time slot set. Thus, the use of overlapping time slots leads improves the quality of the a priori route and time slot assignment, but, in case of the SAA methods, also to higher computation times.

6.5 Non-Anticipation and Ascending Time Slots

So far, we have investigated the performance of LSH, SAA with Fixed A Priori Route and Full SAA. In Section 5, we presented two additional heuristics based on simplifications of the stochastic program: SAA without Non-Anticipation and SAA with Ascending Time Slots. In SAA without Non-Anticipation, the non-anticipatory constraints are relaxed, while in SAA with Ascending Time Slots, constraints requiring the time slots to be ascending along the a priori route are added. Next, we investigate the benefits of these two simplifications when they are incorporated in SAA with Fixed A Priori Route.

Table 6 shows the expected revenue and computation time of SAA without Non-Anticipation and SAA with Ascending Time Slots as a fraction of the corresponding values when using SAA with Fixed A Priori Route, for non-overlapping time slot set (35) and time slot width $w = 0.125T^{\text{TSP}}$, which are the

Table 5: Results for the instances with overlapping time slots.

					Exact Exp. Rev.			Exact Exp. Rev. (Fact. Non-overlap)			CPU time (s)			CPU time (Fact. Non-overlap)		
	n	w	Ov.	T	LSH	SAA Fixed	SAA Full	LSH	SAA Fixed	SAA Full	LSH	SAA Fixed	SAA Full	LSH	SAA Fixed	SAA Full
c60	8	0.25	2	0.90	3.79	3.79	3.81	1.036	1.025	1.011	0.11	354.53	11033.42	0.970	2.127	3.655
		0.25	2	0.75	3.26	3.30	3.35	1.064	1.027	1.019	0.16	838.62	32696.81	0.973	3.123	5.233
		0.25	2	0.60	2.58	2.67	2.78	1.039	1.021	1.010	0.18	171.17	7666.13	1.044	2.855	5.364
c60	8	0.125	2	0.90	3.72	3.73	3.76	1.035	1.028	1.013	0.12	806.40	10819.73	0.967	1.113	2.277
		0.125	2	0.75	3.02	3.24	3.29	1.040	1.017	1.008	0.16	1449.00	31261.28	0.988	1.969	5.863
		0.125	2	0.60	2.35	2.63	2.74	1.033	1.015	1.007	0.17	533.17	5578.24	1.008	2.162	5.178
c60	12	0.25	2	0.90	5.81	5.81		1.046	1.021		2.20	871.08		0.856	2.871	
		0.25	2	0.75	5.06	5.13		1.056	1.035		4.80	4389.65		1.000	6.763	
		0.25	2	0.60	4.05	4.18		1.038	1.033		6.26	1878.74		1.031	5.563	
c60	12	0.125	2	0.90	5.72	5.76		1.050	1.022		2.55	1828.14		0.905	1.408	
		0.125	2	0.75	4.72	5.01		1.030	1.016		4.81	15955.04		1.007	7.237	
		0.125	2	0.60	3.76	4.06		1.026	1.011		5.54	6772.20		0.967	6.813	

more difficult instance sets. We see that SAA without Non-Anticipation for high sample sizes does not affect the solution quality, but that it requires less computation time, for the $n = 8$ customer instance set on average only 21% of the computation time required with customer anticipation. For the $n = 12$ customer instance set, the computation time benefits are mixed. When the horizon is long ($T = 0.9T^{\mathrm{TSP}}$), computation times are significantly smaller, but when the horizon is short ($T = 0.6T^{\mathrm{TSP}}$), the computation time significantly increases. We observe more consistent computational benefits for SAA with Ascending Time Slots, especially for large sample sizes ($N \geq 16$). Interestingly, for small sample sizes ($N \leq 4$) and long horizons ($T = 0.75T^{\mathrm{TSP}}$ and $T = 0.90T^{\mathrm{TSP}}$) enforcing ascending time slots results in an increase in expected revenue. More generally, when the horizon is long, restricting time slots to be ascending along the (fixed TSP) a priori route does not appear to be limiting (in fact, it may offer a benefit). This is likely due to the fact that with a long horizon, most customer arrival sequences lead in their entirety to feasible delivery routes, which, by construction, visit customer locations in the same order as the a priori route.

Table 6: Results of SAA without Non-Anticipation and SAA with Ascending Time Slots.

				Exact Expected Revenue (Factor of Fixed Route SAA)												CPU time (Factor of Fixed Route SAA)											
				Fixed Route SAA without Non-Anticip.						Fixed Route SAA with Asc. Time Slots						Fixed Route SAA without Non-Anticip.						Fixed Route SAA with Asc. Time Slots					
	n	w	T	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$
c60	8	0.125	0.90	1.009	1.003	1.000	1.000	1.000	1.000	1.013	1.002	1.000	1.000	1.000	1.000	0.609	0.514	0.437	0.374	0.296	0.138	0.810	0.820	0.782	0.743	0.646	0.578
		0.125	0.75	1.023	0.993	1.001	0.999	0.999	0.999	1.036	1.002	1.001	1.000	1.000	1.000	0.644	0.559	0.535	0.544	0.429	0.203	0.797	0.797	0.733	0.637	0.640	0.523
		0.125	0.60	1.018	1.007	0.997	0.999	0.999	1.000	1.002	0.995	0.983	0.982	0.982	0.982	0.668	0.616	0.584	0.561	0.505	0.296	0.810	0.779	0.740	0.665	0.577	0.435
c60	12	0.125	0.90	0.995	1.001	1.000	1.000	1.000		1.050	1.001	1.000	1.000	1.000		1.259	1.203	0.959	0.480	0.195		0.890	0.983	0.967	0.833	0.752	
		0.125	0.75	1.019	1.007	1.001	1.000	1.000		1.047	1.014	1.001	1.000	0.999		1.271	1.220	1.096	1.877	0.742		1.004	1.004	0.926	0.704	0.549	
		0.125	0.60	1.005	0.999	0.999	0.999	1.000		1.016	0.990	0.988	0.986	0.986		1.251	1.248	1.290	2.028	2.154		1.076	1.042	0.950	0.719	0.422	

7 Discussion and Future Research Directions

In this paper, we introduce Strategic Time Slot Management, a novel variant of Time Slot Management that simplifies the management of time slots during the ordering process, allows smoothing of fulfillment center operations, and creates delivery consistency. We investigate the design problem for the single-vehicle (or single-route) case. We model the design problem, which involves finding an a priori route and a time slot assignment, as a two-stage stochastic program (where the second stage is an "evaluation" stage modeling the the customer ordering process). We develop a Sample Average Approximation approach for its solution (which uses a highly efficient Dynamic Programming algorithm to evaluate the expected revenue over all possible order placement sequences). An extensive computational study shows the efficacy of the solution approach and provides valuable insights in the benefits of Strategic Time Slot Management.

We see many opportunities for further research into this exciting innovative practice in the online grocery retail sector. We discuss some of these in this final section. Specifically, we consider the extension to the multi-vehicle setting, the aggregation of customer locations, the dynamic management of a priory routes, and the use of Benders Decomposition to improve the efficiency of the Sample Average Approximation approach.

In practice, online grocery retailers employ multiple vehicles to serve their customers. Our two-stage stochastic program can be modified to allow for multiple a priori routes, but it is highly likely that its solution will be more challenging, because the a priori routes need to be balanced across the customer locations. However, a few natural phases solution approaches are worth exploring. Once it has been decided which groups of customer locations to serve using a single a priori route, the problem decomposes into single-vehicle problems for each group of customer locations, and either one of the solution approaches developed in this paper can be applied. From a business perspective, the multi-vehicle case opens up other interesting opportunities. By allowing customer locations to be included in multiple a priori routes, their service can be increased.

In our definition of the Strategic Time Slot Management design problem, we have assumed that the locations that have to be visited represent a delivery address. In practice, it is more likely that the locations that have to be visited represent areas (i.e., groups or clusters of delivery addresses). In such an environment, new design decisions arise, e.g., how to define the areas, and the proposed methodology may have to be revisited. It is possible to adjust the probabilities p_i (and revenues r_i) to reflect knowledge about the customers in an area, but it may be better to consider a different approach, e.g., a discrete probability distribution for the number of order placements in an area. Furthermore, it may no longer be reasonable to assume that the service time at a “location” is constant.

In practice, the set of customers may vary over time (Picnic’s client base is growing rapidly) and the customer characteristics may vary over time (e.g., order placement probability and revenue). This suggest dynamic management of the a priori routes and time slot assignments. However, customer value consistency and may not like to see their assigned time slots change (too) frequently.

Our presented two-stage stochastic program contains many big-M constraints, especially in the second stage, which results in weak LP relaxations and, consequently, large search trees and long solve times. However, as we have seen, given an a priori route and time slot assignment, the second stage is “simply” an evaluation of the expected revenue. Benders decomposition may allow us to solve the integer programs more efficient by exploiting this structure, i.e., incorporate knowledge about the expected costs of a design in the form of optimality cuts. Only further research can tell whether this can lead to computationally viable approaches.

Acknowledgments

This research was funded by NWO, the Netherlands Organisation for Scientific Research, as part of the multi-annual research programme on Sustainable Logistics, and by Stichting Erasmus Trustfonds.

References

N. Agatz, A. M. Campbell, M. Fleischmann, and M. W. P. Savelsbergh. Time slot management in attended home delivery. *Transportation Science*, 45(3):435–449, 2011.

N. Agatz, A. M. Campbell, M. Fleischmann, J. van Nunen, and M. W. P. Savelsbergh. Revenue management opportunities for internet retailers. *Journal of Revenue & Pricing Management*, 12(2):128–138, 2013.

E. Angelelli, C. Archetti, C. Filippi, and M. Vindigni. The probabilistic orienteering problem. *Computers & Operations Research*, 81:269 – 281, 2017.

B. P. Bruck, J.-F. Cordeau, and M. Iori. A practical time slot management and routing problem for attended home services. *Omega*, 81:208 – 219, 2018.

A. M. Campbell and M. W. P. Savelsbergh. Decision support for consumer direct grocery initiatives. *Transportation Science*, 39(3):313–327, 2005.

A. M. Campbell and M. W. P. Savelsbergh. Incentive schemes for attended home delivery services. *Transportation Science*, 40(3):327–341, 2006.

A. M. Campbell and B. W. Thomas. Challenges and advances in a priori routing. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, Operations Research/Computer Science Interfaces, pages 123–142. Springer US, Boston, MA, 2008a.

A. M. Campbell and B. W. Thomas. Probabilistic traveling salesman problem with deadlines. *Transportation Science*, 42(1):1–21, 2008b.

C. Cleophas and J. F. Ehmke. When are deliveries profitable? *Business & Information Systems Engineering*, 6(3): 153–163, 2014.

J. F. Ehmke and A. M. Campbell. Customer acceptance mechanisms for home deliveries in metropolitan areas. *European Journal of Operational Research*, 233(1):193 – 207, 2014.

A. L. Erera, M. Savelsbergh, and E. Uyar. Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks*, 54(4):270–283, 2009.

M. Gendreau, O. Jabali, and W. Rei. Chapter 8: Stochastic vehicle routing problems. In P. Toth and D. Vigo, editors, *Vehicle Routing*, volume 18 of *MOS-SIAM Series on Optimization*, chapter 8, pages 213–239. SIAM - Society for Industrial and Applied Mathematics, Philadelphia, second edition, 2014.

C. Groër, B. Golden, and E. Wasil. The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643, 2009.

F. Hernandez, M. Gendreau, and J.-Y. Potvin. Heuristics for tactical time slot management: a periodic vehicle routing problem view. *International Transactions in Operational Research*, 24(6):1233–1252, 2017.

P. Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(6):929–936, 1988.

R. Klein, M. Neugebauer, D. Ratkovitch, and C. Steinhardt. Differentiated time slot pricing under routing considerations in attended home delivery. *Transportation Science*, 2017. doi: 10.1287/trsc.2017.0738. Published online in Articles in Advance 20 July 2017.

A. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

C. Köhler, J. F. Ehmke, and A. M. Campbell. Flexible time window management for attended home deliveries. *Omega*, 2019. doi: 10.1016/j.omega.2019.01.001. URL http://www.sciencedirect.com/science/article/pii/S030504831830803X.

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

J. Mendoza, C. Guéret, M. Hoskins, H. Lobit, V. Pillac, T. Vidal, and D. Vigo. VRP-REP: a vehicle routing community repository, 06 2014. URL http://vrp-rep.org.

J. Oyola, H. Arntzen, and D. L. Woodruff. The stochastic vehicle routing problem, a literature review, Part II: solution methods. *EURO Journal on Transportation and Logistics*, 6(4):349–388, Dec 2017.

J. Oyola, H. Arntzen, and D. L. Woodruff. The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*, 7(3):193–221, Sep 2018.

N. J. A. Sloane. Sequence A000522. *The On-Line Encyclopedia of Integer Sequences*, 2010. URL https://oeis.org/A000522.

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

R. Spliet and A. F. Gabor. The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, 49 (4):721–731, 2015.

T. Sterling. Startup Picnic runs grocery delivery bus in Dutch online shopping boom. *Reuters*, September 2018. Retrieved from https://www.reuters.com/article/us-netherlands-grocery-internet/startup-picnic-runs-grocery-delivery-bus-in-dutch-online-shopping-boom-idUSKCN1LZ244.

A. Subramanyam and C. E. Gounaris. A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research*, 248(2):384 – 395, 2016.

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

S. A. Voccia, A. M. Campbell, and B. W. Thomas. The probabilistic traveling salesman problem with time windows. *EURO Journal on Transportation and Logistics*, 2(1):89–107, May 2013.

X. Yang, A. K. Strauss, C. S. M. Currie, and R. Eglese. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation Science*, 50(2):473–488, 2016.

A Non-anticipatory Constraints

In this appendix, we present the non-anticipatory constraints of the stochastic programming formulation in Section 4.1. These constraints ensure that the hth arriving customer $\omega(h)$ in scenario ω places an order if and only if it can be feasibly inserted in the current delivery route (containing only placed customers up to $h-1$). We introduce the following additional decision variables: $e_i^{\omega h}$ and $l_i^{\omega h}$ are the earliest arrival time and latest arrival time at customer i, respectively, given the current delivery route after h customer arrivals in scenario ω. In particular, $e_{\omega(h)}^{\omega h}$ ($l_{\omega(h)}^{\omega h}$) are the earliest (latest) arrival times given that current arriving customer $\omega(h)$ is inserted (even if infeasable) in the delivery route. Furthermore, $\tilde{x}_i^{\omega hF}$ and $\tilde{x}_j^{\omega hB}$ denote if forward arc $(i, \omega(h))$ and backward arc $(\omega(h), j)$, respectively, are used when inserting customer $\omega(h)$ in the current delivery route. Note that we cannot use the delivery route variables $x_{ij}^{\omega h}$ since insertion of customer $\omega(h)$ might be infeasable, and thus this customer will not be inserted into the delivery route. Finally, decision variables $w_i^{\omega hF}$ and $w_i^{\omega hB}$ and indicate if there is waiting time at customer i due to the time slot, in case of earliest arrival time (F) and in case of latest arrival time (B), respectively. These are essentially used for booking-keeping. Let $t^{\max}$ be equal to the maximum travel time and let ϵ be the travel time precision.

The non-anticipatory constraints are given by:

$$e_j^{\omega h} \geq \sum_{s\in\mathcal{T}_j} a_s y_{js} \qquad \forall j \in \mathcal{V}_c^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{37}$$

$$e_j^{\omega h} \geq e_i^{\omega h} + t_{ij} - T(1-x_{ij}^{\omega h}) \qquad \forall\,(i,j)\in\mathcal{A}^{\omega h},\, i,j\neq\omega(h),\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{38}$$

$$e_{\omega(h)}^{\omega h} \geq e_i^{\omega h} + t_{i,\omega(h)} - (T+t^{\max})(1-\tilde{x}_i^{\omega h\mathrm{F}}) \qquad \forall\,(i,\omega(h))\in\mathcal{A}^{\omega h}, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{39}$$

$$e_j^{\omega h} \leq \sum_{s\in\mathcal{T}_j} a_s y_{js} + (T+t^{\max})(1-w_j^{\omega h\mathrm{F}}) \qquad \forall j \in \mathcal{V}_c^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{40}$$

$$e_j^{\omega h} \leq e_i^{\omega h} + t_{ij} + T(1+w_j^{\omega h\mathrm{F}} - x_{ij}^{\omega h}) \qquad \forall\,(i,j)\in\mathcal{A}^{\omega h},\, i,j\neq\omega(h),\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{41}$$

$$e_{\omega(h)}^{\omega h} \leq e_i^{\omega h} + t_{i,\omega(h)} + (T+t^{\max})(1+w_{\omega(h)}^{\omega h\mathrm{F}} - \tilde{x}_i^{\omega h\mathrm{F}}) \qquad \forall\,(i,\omega(h))\in\mathcal{A}^{\omega h}, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{42}$$

$$l_i^{\omega h} \leq \sum_{s\in\mathcal{T}_i} b_s y_{is} \qquad \forall i \in \mathcal{V}_c^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{43}$$

$$l_i^{\omega h} \leq l_j^{\omega h} - t_{ij} + T(1-x_{ij}^{\omega h}) \qquad \forall\,(i,j)\in\mathcal{A}^{\omega h},\, i,j\neq\omega(h),\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{44}$$

$$l_{\omega(h)}^{\omega h} \leq l_j^{\omega h} - t_{\omega(h),j} + (T+t^{\max})(1-\tilde{x}_j^{\omega h\mathrm{B}}) \qquad \forall\,(\omega(h),j)\in\mathcal{A}^{\omega h}, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{45}$$

$$l_i^{\omega h} \geq \sum_{s\in\mathcal{T}_j} b_s y_{is} - (T+t^{\max})(1-w_i^{\omega h\mathrm{B}}) \qquad \forall i \in \mathcal{V}_c^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{46}$$

$$l_i^{\omega h} \geq l_j^{\omega h} - t_{ij} - T(1+w_i^{\omega h\mathrm{B}} - x_{ij}^{\omega h}) \qquad \forall\,(i,j)\in\mathcal{A}^{\omega h},\, i,j\neq\omega(h),\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{47}$$

$$l_{\omega(h)}^{\omega h} \geq l_j^{\omega h} - t_{\omega(h),j} - (T+t^{\max})(1+w_{\omega(h)}^{\omega h\mathrm{B}} - \tilde{x}_j^{\omega h\mathrm{B}}) \qquad \forall\,(\omega(h),j)\in\mathcal{A}^{\omega h}, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{48}$$

$$e_o^{\omega h} = a_o = 0,\; l_d^{\omega h} = b_d = T \qquad \forall h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{49}$$

$$\sum_{i\in\mathcal{V}^{\omega h}\setminus\{\omega(h),d\}} \tilde{x}_i^{\omega h\mathrm{F}} = 1 \qquad \forall h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{50}$$

$$\sum_{j\in\mathcal{V}^{\omega h}\setminus\{o,\omega(h)\}} \tilde{x}_j^{\omega h\mathrm{B}} = 1 \qquad \forall h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{51}$$

$$\tilde{x}_i^{\omega h\mathrm{F}} + \tilde{x}_j^{\omega h\mathrm{B}} \leq 1 + x_{ij}^{\omega,h-1} \qquad \forall\,(i,j)\in\mathcal{A}^{\omega,h-1},\, (i,j)\neq(o,d)\,,\, h\in\{2,\ldots,|\omega|\},\, \omega\in\Omega, \tag{52}$$

$$\tilde{x}_o^{\omega h\mathrm{F}} + \tilde{x}_d^{\omega h\mathrm{B}} \leq 2 - \sum_{j\in\mathcal{V}^{\omega,h-1}\setminus\{d\}} x_{oj}^{\omega,h-1} \qquad \forall h\in\{2,\ldots,|\omega|\},\, \omega\in\Omega, \tag{53}$$

$$u_i + 1 \leq u_{\omega(h)} + n(1-\tilde{x}_i^{\omega h\mathrm{F}}) \qquad \forall\,(i,\omega(h))\in\mathcal{A}^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{54}$$

$$u_{\omega(h)} + 1 \leq u_j + n(1-\tilde{x}_j^{\omega h\mathrm{B}}) \qquad \forall\,(\omega(h),j)\in\mathcal{A}^{\omega h},\, h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{55}$$

$$e_{\omega(h)}^{\omega h} \leq l_{\omega(h)}^{\omega h} + 2(T+t^{\max})(1-z_{\omega(h)}^{\omega}) \qquad \forall h\in\{1,\ldots,|\omega|\},\, \omega\in\Omega, \tag{56}$$

$$e_{\omega(h)}^{\omega h} \geq l_{\omega(h)}^{\omega h} + \epsilon - 2(T + t^{\max}) z_{\omega(h)}^{\omega} \qquad \forall h \in \{1, \ldots, |\omega|\}, \omega \in \Omega, \tag{57}$$

$$w_i^{\omega h \mathrm{F}} \in \{0,1\}, w_j^{\omega h \mathrm{B}} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}^{\omega h}, h \in \{1, \ldots, |\omega|\}, \omega \in \Omega, \tag{58}$$

$$\tilde{x}_i^{\omega h \mathrm{F}} \in \{0,1\}, \tilde{x}_j^{\omega h \mathrm{B}} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}^{\omega, h-1}, h \in \{2, \ldots, |\omega|\}, \omega \in \Omega, \tag{59}$$

$$\tilde{x}_o^{\omega 1 \mathrm{F}} = 1, \tilde{x}_d^{\omega 1 \mathrm{B}} = 1 \qquad \forall \omega \in \Omega. \tag{60}$$

Essentially, the first constraints ensure the earliest arrival times and latest arrival times are set exactly. These are obtained by linearizing: $e_j^{\omega h} = \max\left\{\sum_{s \in \mathcal{T}_j} a_s y_{js}, e_i^{\omega h} + t_{ij}\right\}$ in case $x_{ij}^{\omega h} = 1$ and $e_{\omega(h)}^{\omega h} = \max\left\{\sum_{s \in \mathcal{T}_\omega(h)} a_s y_{\omega(h)s}, e_i^{\omega h} + t_{i,\omega(h)}\right\}$ in case $\tilde{x}_i^{\omega hF} = 1$, and $l_i^{\omega h} = \min\left\{\sum_{s \in \mathcal{T}_i} b_s y_{is}, l_j^{\omega h} - t_{ij}\right\}$ in case $x_{ij}^{\omega h} = 1$ and $l_{\omega(h)}^{\omega h} = \min\left\{\sum_{s \in \mathcal{T}_{\omega(h)}} b_s y_{\omega(h)s}, l_j^{\omega h} - t_{\omega(h),j}\right\}$ in case $\tilde{x}_j^{\omega hB} = 1$. Next, the $\tilde{x}_i^{\omega hF}$ and $\tilde{x}_j^{\omega hB}$ decision variables are set correctly. Finally, the feasibility check involves checking if $e_{\omega(h)}^{\omega h} \leq l_{\omega(h)}^{\omega h}$. The revenue $z_{\omega(h)}^{\omega}$ of the hth arriving customer in scenario ω is only obtained if and only if $e_{\omega(h)}^{\omega h} \leq l_{\omega(h)}^{\omega h}$. The time precision ϵ is used to capture the infeasible case: $e_{\omega(h)}^{\omega h} \geq l_{\omega(h)}^{\omega h} + \epsilon$.

B Detailed Results

In this section, we provide some detailed results of our computational study. In the following tables, the left column contains the instance names, which consist of "STSM_c60_", followed by the number of customer locations $n \in \{4, 8, 12\}$, followed by the instance number (1–10), "_Wf" with the time slot width factor ("90", "75", "60"), followed by "o2" in case of the overlapping time slot set, and followed by "_Tf" with the planning horizon factor ("25", "12-5").

Table 7: Detailed results for the Linear Scaling Heuristic, the SAA method with Fixed A Priori Route and the full SAA method, over the $n = 4$ customer instances with non-overlapping set of possible time slots. Bold expected revenues indicate the best known solution.

	Exact Expected Revenue													CPU time (s)												
		Fixed Route SAA						Full SAA							Fixed Route SAA						Full SAA					
	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$
STSM_c60_4_1_Wf25_Tf90	1.6615	1.6615	1.6615	1.6615	1.6615	1.6615	1.6615	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	0.0024	2.8154	0.8644	1.4682	2.5595	5.2293	11.8274	25.2291	1.6237	3.6011	9.8755	31.1044	123.1838
STSM_c60_4_2_Wf25_Tf90	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0031	0.4909	0.6796	1.1927	2.3375	4.2163	9.2481	0.5308	0.6891	2.0809	3.8852	8.3945	41.7038
STSM_c60_4_3_Wf25_Tf90	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0023	0.4273	0.7656	1.2991	2.5274	4.6988	9.7983	0.8247	1.2134	1.8576	7.6939	17.1046	95.0539
STSM_c60_4_4_Wf25_Tf90	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0024	0.4956	0.6977	1.3499	2.3511	4.7108	9.4819	0.5260	0.8053	1.5980	4.6505	13.8634	31.7876
STSM_c60_4_5_Wf25_Tf90	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0021	0.6368	0.7851	1.4786	2.3171	4.5692	8.7096	2.0146	1.5829	5.0816	9.1930	31.4700	91.3528
STSM_c60_4_6_Wf25_Tf90	1.8021	1.8021	1.8021	1.8021	1.8021	1.8021	1.8021	1.8021	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0032	0.5775	0.8206	1.2226	2.1618	5.0714	9.4624	1.5233	1.9320	2.5396	4.2746	29.2446	83.3330
STSM_c60_4_7_Wf25_Tf90	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0022	0.4816	0.7497	1.3151	2.4907	4.6567	9.0390	1.0653	3.4697	4.9168	14.9386	48.4599	132.4651
STSM_c60_4_8_Wf25_Tf90	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	1.7500	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0032	0.5695	0.6900	1.2878	2.3719	4.9947	9.4658	2.0288	1.5289	4.9127	11.1669	41.0009	108.4211
STSM_c60_4_9_Wf25_Tf90	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	0.0026	0.5359	0.7925	1.2765	2.4725	4.2511	9.3801	0.8706	1.1230	2.0323	4.7806	10.0183	50.0919
STSM_c60_4_10_Wf25_Tf90	1.6615	1.7500	1.8021	1.8021	1.8021	1.8021	1.8021	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0022	0.5155	0.8425	1.1304	2.2799	4.3995	9.4262	1.3343	1.7247	2.7198	8.6871	30.7061	111.1003
STSM_c60_4_1_Wf25_Tf75	1.4792	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0041	0.5348	0.8195	1.2399	2.7573	4.7782	10.0782	0.7855	0.9505	2.7119	10.3745	22.4577	86.4923
STSM_c60_4_2_Wf25_Tf75	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0024	0.5078	0.7021	1.1473	2.1543	3.9260	7.9171	0.6082	0.6699	1.1235	2.1235	4.9994	10.0728
STSM_c60_4_3_Wf25_Tf75	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	1.3958	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	0.0031	0.4380	0.7377	1.1834	2.1342	4.2448	9.1078	0.7640	1.4940	2.7186	6.4130	14.8199	63.7753
STSM_c60_4_4_Wf25_Tf75	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0040	0.4202	0.6292	1.1739	2.3418	4.1000	8.3974	0.4971	0.6523	1.5417	2.7402	6.0544	19.2312
STSM_c60_4_5_Wf25_Tf75	1.5625	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0042	3.1403	0.8872	1.2740	2.5439	5.0595	10.4118	64.5953	3.0614	5.2664	11.1210	36.1713	127.5366
STSM_c60_4_6_Wf25_Tf75	1.4792	1.4688	1.4792	1.4792	1.4792	1.4792	1.4792	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0024	0.4787	0.7631	1.2345	2.4812	4.8103	9.5262	0.5882	0.7045	1.4973	3.9339	14.6264	37.1571
STSM_c60_4_7_Wf25_Tf75	1.4688	1.5833	1.5833	1.5833	1.5833	1.5833	1.5833	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	**1.6615**	0.0029	0.5408	0.6867	1.1922	2.4978	4.8892	10.0838	1.2724	1.1611	3.1828	8.4496	28.5784	96.9226
STSM_c60_4_8_Wf25_Tf75	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0027	0.5889	0.7355	1.3146	2.2828	4.8438	9.3381	0.9659	1.3857	3.4231	6.4650	23.8824	64.5649
STSM_c60_4_9_Wf25_Tf75	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0024	1.5800	0.6631	1.2384	2.3909	4.5175	8.7567	8.1236	0.6615	1.2043	2.3138	6.7692	13.2432
STSM_c60_4_10_Wf25_Tf75	1.5625	1.5625	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0033	0.5903	0.7516	1.3969	2.3756	5.1798	9.3854	0.7811	1.4944	2.6549	7.2672	26.2311	67.7513
STSM_c60_4_1_Wf25_Tf60	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	0.0033	0.5032	0.7752	1.0546	2.2310	4.5086	9.0551	1.1184	1.9572	1.5741	4.8055	13.5819	27.3861
STSM_c60_4_2_Wf25_Tf60	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	0.0024	0.4658	0.6333	0.9596	1.8514	3.5886	7.6681	0.5865	0.6402	1.0163	1.8270	3.5581	7.7978
STSM_c60_4_3_Wf25_Tf60	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	0.0025	0.4593	0.5674	1.1517	2.1471	3.9949	8.2498	0.5151	0.5274	1.1016	2.0517	3.8511	8.2685
STSM_c60_4_4_Wf25_Tf60	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0025	0.5309	0.5719	1.1075	2.3156	4.1835	7.8556	0.5915	0.5800	1.0915	2.3628	4.3407	8.9472
STSM_c60_4_5_Wf25_Tf60	1.1146	1.1146	1.1146	1.1146	1.1146	1.1146	1.1146	**1.2552**	**1.2552**	**1.2552**	**1.2552**	**1.2552**	**1.2552**	0.0028	0.4225	0.6150	1.3255	2.2888	4.5850	8.8924	0.5125	0.6036	1.1124	2.2844	4.5782	10.8052
STSM_c60_4_6_Wf25_Tf60	1.2552	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	1.2552	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	0.0025	0.4774	0.6615	1.1647	2.3665	4.0090	8.6591	0.4548	1.0170	2.3503	5.1282	10.6993	24.0093
STSM_c60_4_7_Wf25_Tf60	1.3750	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	**1.4792**	1.3958	**1.4792**	**1.4792**	**1.4792**	**1.4792**	0.0027	0.6103	0.6621	1.2510	2.1507	4.7050	9.4230	0.8658	0.8959	2.4721	4.6770	14.9027	44.5551
STSM_c60_4_8_Wf25_Tf60	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	0.0027	0.5264	0.6921	1.2027	2.2351	4.3561	8.7167	0.9045	0.9145	1.5610	3.7322	9.9562	28.0523
STSM_c60_4_9_Wf25_Tf60	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0023	0.4485	0.6318	1.2761	2.0957	4.1100	8.4794	0.4328	0.6282	1.3166	2.4432	4.6381	12.5873
STSM_c60_4_10_Wf25_Tf60	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	0.0026	0.4909	0.6441	1.3433	2.4315	4.5552	8.5428	0.5558	0.8455	1.6513	3.6496	8.9288	19.3212
STSM_c60_4_1_Wf12-5_Tf90	1.6615	1.6615	1.6615	1.6615	1.6615	1.6615	1.6615	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	0.0024	7.5700	0.9299	1.5761	3.3393	10.9902	38.4730	86.2555	1.6130	3.6359	16.1260	55.6250	247.7796
STSM_c60_4_2_Wf12-5_Tf90	1.3750	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0032	0.4582	0.8574	1.4453	2.5554	4.9989	10.8290	0.4920	1.0970	1.3952	4.0245	11.6819	43.5793
STSM_c60_4_3_Wf12-5_Tf90	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	0.0023	0.5048	0.9131	1.6266	2.9285	5.8763	17.4688	0.6895	1.8726	4.2310	11.9103	53.4223	280.7118
STSM_c60_4_4_Wf12-5_Tf90	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0036	0.5659	0.8316	1.6485	2.7910	5.8778	11.9474	0.5474	1.1867	2.0411	4.1659	15.0265	46.7993
STSM_c60_4_5_Wf12-5_Tf90	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	0.0027	0.5804	0.8738	1.7252	3.0778	8.6712	22.3905	1.2922	2.1237	5.0661	18.3024	81.5265	421.8939
STSM_c60_4_6_Wf12-5_Tf90	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	**1.8021**	1.7500	**1.8021**	**1.8021**	**1.8021**	**1.8021**	0.0024	5.6047	0.8421	1.3051	3.0585	7.1254	25.6569	44.5425	1.1427	2.3842	10.8330	57.2006	238.5430
STSM_c60_4_7_Wf12-5_Tf90	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	**1.8750**	**1.8750**	**1.8750**	**1.8750**	**1.8750**	0.0023	0.4848	0.9489	1.6744	3.0593	7.0180	23.2387	0.4630	4.4672	4.1142	11.6297	54.0899	186.4464
STSM_c60_4_8_Wf12-5_Tf90	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	**1.7500**	0.0022	0.5025	0.9682	1.7258	3.1715	6.4608	26.3827	2.1051	4.3455	7.5056	15.0714	66.0987	444.9649
STSM_c60_4_9_Wf12-5_Tf90	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0025	0.5311	0.8264	1.6370	2.7554	5.8854	20.4292	0.5380	0.9501	2.1475	7.6439	25.5300	119.6919
STSM_c60_4_10_Wf12-5_Tf90	1.5000	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	**1.8021**	**1.8021**	**1.8021**	**1.8021**	0.0024	0.6351	0.9103	1.5991	2.9580	6.9573	22.7811	1.1028	1.1064	4.8624	14.3005	70.0406	310.7989
STSM_c60_4_1_Wf12-5_Tf75	1.3958	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0040	0.4688	0.9131	1.5397	2.8105	5.9529	26.0239	0.5063	1.0347	2.8046	7.1306	25.3897	132.7182
STSM_c60_4_2_Wf12-5_Tf75	1.2552	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0025	0.4389	0.7586	1.2524	2.2051	4.6475	9.7199	0.5019	0.6765	1.4352	2.0927	5.7087	20.3300
STSM_c60_4_3_Wf12-5_Tf75	1.3750	**1.3958**	1.3750	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	0.0025	0.4947	0.7691	1.4590	2.4011	4.6586	10.8821	0.6158	0.7406	1.7315	3.8623	13.3538	61.0304
STSM_c60_4_4_Wf12-5_Tf75	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0039	0.4942	0.8202	1.5609	2.4940	4.8821	10.0908	0.6489	0.8866	2.0406	3.0190	7.9366	35.4175
STSM_c60_4_5_Wf12-5_Tf75	1.3958	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0031	0.5205	0.9002	1.4825	2.8260	5.7130	28.5125	0.6399	1.1545	2.5329	8.3855	26.9210	189.2329
STSM_c60_4_6_Wf12-5_Tf75	1.3594	1.4688	1.4688	1.4688	1.4688	1.4688	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0025	0.5187	0.7520	1.2792	2.7095	5.0596	10.4021	0.5222	0.7772	1.3208	3.1917	10.6988	43.7553
STSM_c60_4_7_Wf12-5_Tf75	1.2552	1.5833	1.5833	1.5833	1.5833	1.5833	1.5833	1.5833	1.5833	**1.6615**	**1.6615**	**1.6615**	**1.6615**	0.0027	0.5966	0.8252	1.5485	2.9704	5.4037	21.6392	1.1146	0.9836	3.1228	10.7572	24.2798	143.0879
STSM_c60_4_8_Wf12-5_Tf75	1.3750	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	1.4688	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0026	0.5335	0.7825	1.5225	3.0910	5.5869	23.9201	0.7562	1.2579	3.4717	11.1544	29.1031	143.6727
STSM_c60_4_9_Wf12-5_Tf75	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0024	0.5331	0.8147	1.3303	2.4131	5.0020	10.8423	0.6541	0.7813	1.4196	2.3552	5.3715	22.3695
STSM_c60_4_10_Wf12-5_Tf75	1.3958	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	**1.5833**	0.0030	0.5449	0.7710	1.4305	2.5543	5.0622	22.6646	0.6690	0.7951	2.0864	5.0258	17.4979	134.3897
STSM_c60_4_1_Wf12-5_Tf60	0.9375	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	0.0028	0.5139	0.7990	1.4122	2.4842	4.8346	11.3654	0.6427	0.8571	1.4942	3.1194	9.0814	44.7025
STSM_c60_4_2_Wf12-5_Tf60	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	**0.8750**	0.0023	0.4078	0.5742	1.1574	1.9290	4.0797	8.0723	0.4940	0.5631	1.0847	1.8122	3.8803	7.6118
STSM_c60_4_3_Wf12-5_Tf60	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	**1.1146**	0.0026	0.5521	0.7569	1.1221	2.2552	4.5229	9.4198	0.6480	0.6971	1.0485	2.1206	4.2972	8.7764
STSM_c60_4_4_Wf12-5_Tf60	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0033	0.4945	0.6194	1.1077	2.0981	4.2030	8.3652	0.5886	0.5950	1.1033	2.0456	4.3137	8.9354
STSM_c60_4_5_Wf12-5_Tf60	0.9375	1.1146	1.1146	1.1146	1.1146	1.1146	1.1146	**1.2552**	**1.2552**	**1.2552**	**1.2552**	**1.2552**	**1.2552**	0.0028	0.4771	0.7513	1.2902	2.2940	4.8291	11.0497	0.5610	0.6626	1.2262	2.6916	5.6144	16.2488
STSM_c60_4_6_Wf12-5_Tf60	0.9375	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	1.2552	**1.3594**	**1.3594**	**1.3594**	**1.3594**	**1.3594**	0.0028	0.4840	0.7366	1.1281	2.1817	4.2876	9.1154	0.6192	0.7783	1.2534	2.5460	6.2173	17.2593
STSM_c60_4_7_Wf12-5_Tf60	1.3750	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	**1.3958**	0.0026	0.5612	0.7611	1.3799	2.4012	5.2520	9.9638	0.6873	0.8685	1.8031	4.4869	13.0605	43.7891
STSM_c60_4_8_Wf12-5_Tf60	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	1.2552	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	0.0025	0.4950	0.7018	1.3191	2.5396	5.0070	11.2721	0.6222	0.7712	1.5253	3.2125	7.6117	30.7989
STSM_c60_4_9_Wf12-5_Tf60	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	**1.4688**	0.0024	0.4412	0.6848	1.1355	2.0758	4.1465	8.8745	0.5738	0.6717	1.1172	2.2641	4.6802	10.8011
STSM_c60_4_10_Wf12-5_Tf60	0.9375	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	1.2552	**1.3750**	**1.3750**	**1.3750**	**1.3750**	**1.3750**	0.0042	0.5429	0.6667	1.4579	2.6234	4.4167	11.3971	0.6061	0.6779	1.3233	2.8905	5.5714	33.6590

Table 8: Detailed results for the Linear Scaling Heuristic, the SAA method with Fixed A Priori Route and the full SAA method, over the $n = 8$ customer instances with non-overlapping set of possible time slots. Bold expected revenues indicate the best known solution.

	Exact Expected Revenue											CPU time (s)										
		Fixed Route SAA						Full SAA					Fixed Route SAA						Full SAA			
	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$
STSM_c60_8_1_Wf25_Tf90	**3.9031**	**3.9031**	**3.9031**	**3.9031**	**3.9031**	**3.9031**	**3.9031**	3.6094	**3.9031**	**3.9031**	**3.9031**	0.0950	3.4415	4.4383	6.6333	11.4799	21.9048	43.9930	16.5561	40.0104	195.6657	1776.0971
STSM_c60_8_2_Wf25_Tf90	3.5232	3.3281	**3.7271**	**3.7271**	**3.7271**	**3.7271**	**3.7271**	3.3739	3.6094	**3.7271**	**3.7271**	0.1154	3.9636	4.8255	8.5346	15.1091	27.2425	88.3367	40.3693	121.2001	738.6884	2133.1758
STSM_c60_8_3_Wf25_Tf90	3.7822	3.8496	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8309	**3.8752**	**3.8752**	0.1033	3.8968	4.8266	6.9839	14.2025	26.8991	56.1909	41.9264	38.4854	260.9196	2500.5682
STSM_c60_8_4_Wf25_Tf90	3.7526	3.7526	3.7526	3.7526	3.7526	3.7526	3.7526	3.7526	3.7652	3.7897	**3.8685**	0.1144	4.0004	5.2342	9.0758	16.5579	38.5290	96.4141	35.7914	239.3991	961.4871	3188.7477
STSM_c60_8_5_Wf25_Tf90	3.5997	3.6094	3.6094	3.6094	3.6094	3.6094	3.6094	3.6623	3.7718	**3.8032**	3.7941	0.1232	4.1491	5.5844	9.5241	16.8424	60.5236	287.1675	62.9295	102.9088	5549.5045	4453.8667
STSM_c60_8_6_Wf25_Tf90	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.4688	3.6723	3.7790	**3.7986**	0.1101	4.4451	5.9422	9.3176	16.6980	35.5423	213.7624	83.6241	159.1652	663.2798	3206.9923
STSM_c60_8_7_Wf25_Tf90	3.7114	3.7114	3.7114	3.7114	3.7114	3.7114	3.7114	3.6466	3.5670	3.7114	**3.7443**	0.1164	4.6133	5.6425	9.9454	15.0967	36.7025	177.9607	49.2442	131.1813	1117.8503	2862.7628
STSM_c60_8_8_Wf25_Tf90	3.6004	3.1945	3.5892	3.6535	3.6535	3.6535	3.6535	3.1605	3.5901	**3.7283**	**3.7283**	0.1178	4.7082	5.7314	8.3310	16.1865	53.6368	223.8823	74.1009	115.0879	387.7500	2036.1914
STSM_c60_8_9_Wf25_Tf90	3.4514	3.3544	3.4847	3.4847	3.4847	3.4847	3.4847	3.0694	3.4052	3.5549	**3.5738**	0.1350	4.6519	6.6524	9.0949	29.6796	64.3446	280.8740	46.8127	242.0769	405.7612	5471.5734
STSM_c60_8_10_Wf25_Tf90	3.6417	3.3404	3.6417	3.6417	3.6417	3.6417	3.6417	3.3404	3.4688	**3.6417**	**3.6417**	0.1354	4.5454	5.5988	8.3403	15.5081	42.3359	198.0217	34.9472	73.2097	369.0437	2559.2699
STSM_c60_8_1_Wf25_Tf75	3.1775	3.2958	3.2044	3.2958	3.2958	3.2958	3.2958	3.1589	3.3785	3.3911	**3.4141**	0.1559	4.5643	6.4295	8.9811	19.9606	61.6603	278.5116	48.5438	167.0430	569.2882	4027.0389
STSM_c60_8_2_Wf25_Tf75	2.8570	3.3652	3.3875	3.3875	3.3875	3.3875	3.3875	3.3652	**3.4040**	3.3467	**3.4040**	0.1779	4.2026	5.5323	8.1894	14.8325	29.0533	69.6919	28.3048	100.2021	233.0675	1489.8695
STSM_c60_8_3_Wf25_Tf75	3.0509	3.2067	3.2559	3.2559	3.2559	3.2559	3.2559	3.1995	3.3404	**3.3661**	**3.3661**	0.1624	4.6289	5.7173	10.1604	21.5367	45.3234	310.1643	124.1529	404.1429	555.8855	9126.9930
STSM_c60_8_4_Wf25_Tf75	3.3846	3.3945	3.3846	3.3958	3.3958	3.3958	3.3958	3.2168	3.2733	**3.4013**	3.3958	0.1438	4.6547	5.8701	11.6729	20.6831	49.0279	225.9224	93.1554	239.1346	2976.6316	9096.4595
STSM_c60_8_5_Wf25_Tf75	3.2940	3.3700	3.3700	3.3700	3.3700	3.3700	3.3700	3.3297	3.2392	3.3752	**3.3940**	0.1389	4.5104	6.5466	9.9706	20.6422	60.1048	234.4747	37.0950	227.4143	709.7707	5723.8579
STSM_c60_8_6_Wf25_Tf75	3.1957	3.1957	3.2209	3.2209	3.2209	3.2209	3.2209	3.1946	3.1247	3.2684	**3.3062**	0.1598	4.7442	6.4841	10.4842	24.7503	84.8021	293.6710	64.6147	101.4237	609.6594	6224.1714
STSM_c60_8_7_Wf25_Tf75	3.0415	3.0624	3.1768	**3.1985**	**3.1985**	**3.1985**	**3.1985**	2.9704	3.1768	**3.1985**	**3.1985**	0.1639	4.4341	6.2788	9.5096	22.7311	50.1757	259.0523	25.3499	89.5528	240.9273	5098.2877
STSM_c60_8_8_Wf25_Tf75	2.8794	2.9951	2.9951	2.9951	2.9951	2.9951	2.9951	2.8679	3.1023	3.0417	**3.1036**	0.1699	4.8698	6.9787	10.6423	24.6122	67.0397	338.4555	71.1735	209.9303	651.8929	6062.6008
STSM_c60_8_9_Wf25_Tf75	3.0821	3.0365	3.0365	3.0821	3.0821	3.0821	3.0821	2.9832	3.0972	3.1752	**3.2005**	0.1574	4.6310	6.7447	12.2971	23.2291	64.0581	376.0803	48.5877	120.3312	598.5351	4319.1691
STSM_c60_8_10_Wf25_Tf75	2.6537	2.9700	2.9700	2.9700	2.9700	2.9700	2.9700	2.7427	2.8546	**3.0808**	**3.0808**	0.2104	5.2452	7.3652	12.5547	28.8211	80.9024	299.0979	86.4154	163.1040	946.4163	11318.3486
STSM_c60_8_1_Wf25_Tf60	2.8176	2.8480	2.8480	2.8480	2.8480	2.8480	2.8480	2.9334	2.8792	2.9868	**2.9875**	0.1651	4.8108	6.3189	8.3898	15.6796	29.7538	64.2492	33.1143	103.8569	261.8309	1634.7490
STSM_c60_8_2_Wf25_Tf60	2.2488	2.8836	2.8836	2.8836	2.8836	2.8836	2.8836	2.8836	2.8836	**2.9725**	**2.9725**	0.1892	4.8524	5.5318	8.4785	13.6876	25.8520	52.5835	17.6806	32.3701	147.7220	665.8355
STSM_c60_8_3_Wf25_Tf60	2.6838	2.6838	2.7082	2.7082	2.7082	2.7082	2.7082	2.4965	2.7576	2.8226	**2.8566**	0.1604	4.8657	5.8194	9.2240	14.8513	27.1714	59.5624	29.5206	87.6572	325.1373	2143.9151
STSM_c60_8_4_Wf25_Tf60	2.6894	2.7835	2.7835	2.8140	2.8140	2.8140	2.8140	2.7537	2.8933	**2.9227**	**2.9227**	0.1662	4.6648	5.9374	8.7172	15.1675	25.8052	55.7624	111.7284	96.4924	359.6765	3835.2491
STSM_c60_8_5_Wf25_Tf60	2.8006	2.8006	2.7843	2.7843	2.8006	2.8006	2.8006	2.7422	2.8791	**2.9738**	2.9677	0.1440	4.6440	5.9317	9.3655	13.8207	26.5754	68.1286	34.9557	104.3251	465.2014	1379.4820
STSM_c60_8_6_Wf25_Tf60	2.5238	2.6614	2.7231	2.7231	2.7231	2.7231	2.7231	2.5813	2.7251	2.8292	**2.8437**	0.1647	5.1073	6.3265	8.4686	14.3497	25.2397	54.8248	102.2259	89.8386	153.5815	968.9124
STSM_c60_8_7_Wf25_Tf60	2.4110	2.4853	2.4491	2.4853	2.4853	2.4853	2.4853	2.4298	2.5578	2.6669	**2.6914**	0.1869	5.3267	6.5883	8.8033	16.1161	37.6671	90.3331	32.7989	68.4255	194.7554	1194.7916
STSM_c60_8_8_Wf25_Tf60	2.3610	2.4214	2.5111	2.5111	2.5111	2.5111	2.5111	2.3847	2.5787	**2.6293**	**2.6293**	0.1842	4.9680	6.0706	8.6449	15.0399	25.7868	56.5473	49.6963	68.3227	175.6209	1054.3839
STSM_c60_8_9_Wf25_Tf60	2.1964	2.0017	2.1855	2.1964	2.1964	2.1964	2.1964	2.2078	2.3875	**2.3901**	2.3875	0.1691	4.9952	6.2922	9.4278	13.9326	24.8817	49.5870	21.8400	44.7094	160.8514	654.4509
STSM_c60_8_10_Wf25_Tf60	2.1178	2.1366	2.1318	2.1527	2.1527	2.1527	2.1527	2.0281	2.2081	2.2502	**2.2523**	0.1963	5.3146	6.3208	9.3621	14.8717	25.2167	47.8552	17.1185	47.9426	189.8527	759.5215
STSM_c60_8_1_Wf12-5_Tf90	**3.7615**	3.7500	3.7500	**3.7615**	**3.7615**	**3.7615**	**3.7615**	3.3281	3.7500	3.6952	**3.7615**	0.1040	3.9563	4.9225	8.7993	18.5688	54.8438	538.7934	40.1630	26.5288	415.2341	4592.2336
STSM_c60_8_2_Wf12-5_Tf90	3.3856	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	**3.7271**	3.7081	**3.7271**	**3.7271**	0.1318	4.2028	6.4548	11.0658	28.1497	110.0208	1037.2753	28.8422	150.1300	791.0801	4334.6335
STSM_c60_8_3_Wf12-5_Tf90	3.7822	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.4688	3.8750	3.8750	**3.8752**	0.0971	3.5378	5.0760	8.0878	17.2512	36.2496	128.7391	23.0065	87.9150	185.2899	1141.4483
STSM_c60_8_4_Wf12-5_Tf90	3.7526	3.7500	3.7526	3.7526	3.7526	3.7526	3.7526	3.7500	3.6737	3.7442	**3.8138**	0.1152	3.9937	5.9574	9.9091	19.1638	50.2655	336.4036	26.1002	290.5343	1025.1359	3275.3905
STSM_c60_8_5_Wf12-5_Tf90	3.5061	3.5390	3.6094	3.6094	3.6094	3.6094	3.6094	3.5390	3.7500	3.7500	**3.7567**	0.1249	4.4999	5.8132	11.9400	21.4197	72.4673	769.1475	54.6559	67.8557	1417.8360	5317.1676
STSM_c60_8_6_Wf12-5_Tf90	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.4421	3.6723	**3.7318**	**3.7318**	0.1062	4.3968	5.2746	12.2487	23.8709	74.8885	729.5053	25.6565	61.3926	817.3733	5875.7588
STSM_c60_8_7_Wf12-5_Tf90	3.6265	3.4583	3.6309	3.6265	3.6309	3.6309	3.6309	3.1773	3.6106	**3.6999**	**3.6999**	0.1314	4.4428	6.1455	11.6838	20.8904	68.3698	656.3330	17.6250	107.0085	409.3292	2688.8927
STSM_c60_8_8_Wf12-5_Tf90	3.5862	3.5044	3.5436	3.6016	3.6016	3.6016	3.6016	3.3120	3.5862	**3.6330**	**3.6330**	0.1228	4.3163	6.6641	10.3772	23.3863	98.0254	956.7291	14.7194	250.3772	346.3732	7843.6362
STSM_c60_8_9_Wf12-5_Tf90	3.3914	3.3287	3.3557	3.3914	3.3914	3.3914	3.3914	3.2917	3.4784	3.5370	**3.5563**	0.1527	5.4617	7.4493	14.8303	37.4131	145.0833	1129.6980	152.9094	174.8461	1345.9369	10790.7560
STSM_c60_8_10_Wf12-5_Tf90	3.4860	3.3531	3.5177	3.5177	3.5177	3.5177	3.5177	3.3101	3.4736	3.5773	**3.6038**	0.1349	4.8348	6.3029	13.4438	22.1509	108.3703	959.9330	28.9909	85.8991	652.3539	1664.6751
STSM_c60_8_1_Wf12-5_Tf75	2.9789	3.0521	3.2658	3.2658	3.2658	3.2658	3.2658	2.7122	3.3087	**3.4141**	**3.4141**	0.1688	4.7476	7.0400	11.7453	23.3689	71.2499	581.9934	20.0721	182.5327	517.8883	4804.9619
STSM_c60_8_2_Wf12-5_Tf75	2.7283	3.3281	3.3875	3.3875	3.3875	3.3875	3.3875	3.3805	3.3875	**3.4222**	3.4146	0.1461	4.3266	6.0447	10.4924	19.1942	36.9817	428.6065	37.2190	63.2814	319.2527	1893.4117
STSM_c60_8_3_Wf12-5_Tf75	3.0278	2.9816	3.2552	3.2552	3.2552	3.2552	3.2552	3.1048	3.1381	**3.3661**	**3.3661**	0.1480	4.9671	6.3671	10.6689	24.4786	71.6143	752.2795	38.8600	101.2073	588.4222	7958.0495
STSM_c60_8_4_Wf12-5_Tf75	3.1502	3.3945	**3.3958**	**3.3958**	**3.3958**	**3.3958**	**3.3958**	3.3574	3.3653	**3.3958**	**3.3958**	0.1442	4.7602	7.1971	11.0376	19.2292	55.1553	667.6586	42.2215	317.8649	1180.1973	6034.6153
STSM_c60_8_5_Wf12-5_Tf75	3.2746	3.1849	3.3404	3.3019	3.3404	3.3404	3.3404	3.0621	3.3404	3.3031	**3.3923**	0.1224	4.7135	5.8769	13.2683	27.2301	104.3487	874.4288	20.6508	115.8654	1048.9828	11314.5908
STSM_c60_8_6_Wf12-5_Tf75	2.7507	3.0917	3.0969	3.1224	3.1273	3.1273	3.1273	2.7701	3.1573	3.2419	**3.2872**	0.1656	5.2662	7.2626	12.1276	25.5297	102.1824	762.5815	52.1618	120.9413	767.1049	4240.4143
STSM_c60_8_7_Wf12-5_Tf75	2.8231	3.0624	**3.1773**	**3.1773**	**3.1773**	**3.1773**	**3.1773**	2.9664	3.1096	**3.1773**	**3.1773**	0.1702	4.7757	7.0377	10.8474	28.3924	52.9676	618.6154	24.9892	105.7839	329.2869	4675.8158
STSM_c60_8_8_Wf12-5_Tf75	2.8222	2.6844	2.8682	2.8849	2.8849	2.8849	2.8849	2.7480	2.9210	**3.0812**	**3.0812**	0.1543	4.9005	6.7596	10.8923	32.0799	85.1768	837.5822	20.6083	87.1496	352.8540	4906.6893
STSM_c60_8_9_Wf12-5_Tf75	2.8966	3.0365	3.0365	3.0406	3.0406	3.0406	3.0406	3.0365	**3.1469**	**3.1469**	**3.1469**	0.1541	4.6107	6.7199	13.5884	28.5326	93.9260	995.7139	19.4317	69.5771	287.7850	2085.8885
STSM_c60_8_10_Wf12-5_Tf75	2.5620	2.7354	2.9374	2.9374	2.9374	2.9374	2.9374	2.5495	**3.0035**	2.9763	**3.0035**	0.2049	5.0259	6.9042	13.1637	33.5195	100.2246	838.3208	13.9019	107.4354	451.9738	5407.8218
STSM_c60_8_1_Wf12-5_Tf60	2.7750	2.7509	2.8176	2.8238	2.8238	2.8238	2.8238	2.9218	2.8426	**2.9875**	2.9825	0.1364	4.9115	6.5530	10.1542	20.2583	41.0107	264.6366	34.5398	90.8302	214.4942	1724.5450
STSM_c60_8_2_Wf12-5_Tf60	1.8749	2.8836	2.8836	2.8836	2.8836	2.8836	2.8836	2.9055	2.8397	**2.9725**	**2.9725**	0.1927	4.9751	6.7795	10.0193	17.8106	44.6785	246.1209	20.2827	38.9297	98.3588	486.6320
STSM_c60_8_3_Wf12-5_Tf60	2.4849	2.6725	2.7082	2.7082	2.7082	2.7082	2.7082	2.7514	2.8065	2.8226	**2.8566**	0.1633	5.0570	6.0915	10.6632	18.0639	53.4839	224.0327	34.3140	44.6249	244.1074	760.5341
STSM_c60_8_4_Wf12-5_Tf60	2.5780	2.7600	2.8140	2.8140	2.8140	2.8140	2.8140	2.5822	2.8404	2.9031	**2.9291**	0.1518	5.2500	7.2289	10.4625	19.3675	46.0957	405.9890	28.2953	85.5509	335.2479	2448.3325
STSM_c60_8_5_Wf12-5_Tf60	2.7073	2.6321	2.7709	2.7709	2.7709	2.7709	2.7709	2.6869	**2.8693**	**2.8693**	**2.8693**	0.1597	5.6231	7.2911	10.9253	20.6142	54.7886	232.4117	67.2963	138.6443	267.0550	1995.8084
STSM_c60_8_6_Wf12-5_Tf60	2.3956	2.5702	2.6319	2.6385	2.6385	2.6385	2.6385	2.5014	2.7487	2.8118	**2.8292**	0.1818	5.0186	7.3724	9.9299	19.1178	44.2544	306.2187	16.5422	71.7356	81.8062	527.9710
STSM_c60_8_7_Wf12-5_Tf60	2.0461	2.3785	2.4072	2.4517	2.4517	2.4517	2.4517	2.5965	2.5607	**2.6323**	**2.6323**	0.1914	5.8367	7.2842	12.0329	23.6500	58.8941	382.9372	19.4411	57.2712	94.8675	1716.7317
STSM_c60_8_8_Wf12-5_Tf60	2.0954	2.3183	2.4788	2.4788	2.4788	2.4788	2.4788	2.3949	2.3194	2.5093	**2.5374**	0.1632	4.8633	6.2641	9.3935	17.5570	46.0856	227.0039	10.4070	29.2446	79.8245	582.9247
STSM_c60_8_9_Wf12-5_Tf60	2.0771	2.1931	1.9527	2.1931	2.1931	2.1931	2.1931	**2.3875**	2.1798	**2.3875**	**2.3875**	0.1730	4.8156	6.4007	9.5061	15.9667	34.9632	102.2418	11.5050	13.4315	47.5519	197.7013
STSM_c60_8_10_Wf12-5_Tf60	1.7699	2.1178	2.1021	2.1178	2.1366	2.1366	2.1366	1.9844	**2.2506**	**2.2506**	**2.2506**	0.2167	5.5228	6.0753	10.3477	15.9671	31.1015	74.4960	12.2588	10.7288	92.3193	332.1204

Table 9: Detailed results for the Linear Scaling Heuristic and the SAA method with Fixed A Priori Route, over the $n = 12$ customer instances with non-overlapping set of possible time slots. Bold expected revenues indicate the best known solution.

	Exact Expected Revenue						CPU time (s)					
		Fixed Route SAA						Fixed Route SAA				
	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	LSH	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$
STSM_c60_12_1_Wf25_Tf90	5.5843	5.6822	**5.8249**	**5.8249**	**5.8249**	**5.8249**	2.1853	54.0131	46.0148	53.2191	70.1212	106.7082
STSM_c60_12_2_Wf25_Tf90	5.4224	5.3321	5.5830	**5.6157**	**5.6157**	**5.6157**	2.3778	66.7636	62.5304	69.7368	112.8079	390.8039
STSM_c60_12_3_Wf25_Tf90	5.6583	5.6422	**5.7727**	**5.7727**	**5.7727**	**5.7727**	2.3212	63.3391	54.9321	62.2141	160.2439	196.5460
STSM_c60_12_4_Wf25_Tf90	5.1805	5.4543	**5.6442**	**5.6442**	**5.6442**	**5.6442**	2.7320	68.5092	60.2900	68.2493	113.2490	524.9806
STSM_c60_12_5_Wf25_Tf90	**5.8490**	5.2299	**5.8490**	**5.8490**	**5.8490**	**5.8490**	2.3378	65.0484	51.2255	57.0308	79.7275	146.3618
STSM_c60_12_6_Wf25_Tf90	**5.6852**	5.5991	**5.6852**	**5.6852**	**5.6852**	**5.6852**	2.4116	54.4269	55.4539	61.6628	80.4407	219.9455
STSM_c60_12_7_Wf25_Tf90	5.5325	5.6277	**5.7689**	**5.7689**	**5.7689**	**5.7689**	2.4971	65.8473	57.1857	59.4171	89.4279	237.7508
STSM_c60_12_8_Wf25_Tf90	**5.4736**	**5.4736**	**5.4736**	**5.4736**	**5.4736**	**5.4736**	3.2367	71.4775	75.2617	88.4082	161.0514	540.4247
STSM_c60_12_9_Wf25_Tf90	**5.5121**	5.3137	5.4468	**5.5121**	**5.5121**	**5.5121**	3.3342	67.3734	64.5633	74.7631	122.7309	561.0369
STSM_c60_12_10_Wf25_Tf90	5.6938	5.6179	**5.7926**	**5.7926**	**5.7926**	**5.7926**	2.3168	61.2263	49.7309	56.3057	70.7395	109.3079
STSM_c60_12_1_Wf25_Tf75	5.1949	4.8283	**5.2049**	**5.2049**	**5.2049**	**5.2049**	4.4816	82.9824	85.9124	106.8819	180.5659	471.3842
STSM_c60_12_2_Wf25_Tf75	4.8804	4.8055	**5.1626**	**5.1626**	**5.1626**	**5.1626**	3.6263	93.3897	86.7213	117.3851	220.9816	851.7633
STSM_c60_12_3_Wf25_Tf75	4.9778	5.1488	5.1817	5.2596	**5.2650**	**5.2650**	4.4646	78.3220	62.8467	82.3130	176.1439	452.6085
STSM_c60_12_4_Wf25_Tf75	5.1404	5.0765	**5.3122**	**5.3122**	**5.3122**	**5.3122**	3.5138	72.2123	72.0937	85.8769	127.5098	500.1635
STSM_c60_12_5_Wf25_Tf75	4.5680	4.5558	4.5680	**4.5785**	4.5558	**4.5785**	5.3836	96.8942	107.9190	143.2834	295.0963	953.2172
STSM_c60_12_6_Wf25_Tf75	4.2664	**4.3203**	4.2771	**4.3203**	**4.3203**	**4.3203**	5.8707	96.3667	117.1630	144.9426	229.2006	699.3830
STSM_c60_12_7_Wf25_Tf75	**4.7278**	4.4259	4.6977	**4.7278**	**4.7278**	**4.7278**	5.6674	98.2860	104.3954	131.8186	229.0817	783.5195
STSM_c60_12_8_Wf25_Tf75	4.3223	4.5276	**4.6622**	**4.6622**	**4.6622**	**4.6622**	5.7252	88.2139	100.4725	129.8239	233.9715	800.7313
STSM_c60_12_9_Wf25_Tf75	4.6654	5.1495	**5.1868**	**5.1868**	**5.1868**	**5.1868**	5.4438	80.9890	66.7838	80.8778	152.4848	384.7232
STSM_c60_12_10_Wf25_Tf75	**5.1418**	**5.1418**	**5.1418**	**5.1418**	**5.1418**	**5.1418**	3.8209	79.7024	83.2738	99.1277	184.3195	592.8886
STSM_c60_12_1_Wf25_Tf60	4.6521	**4.7447**	**4.7447**	**4.7447**	4.7394	**4.7447**	3.9741	82.9817	83.8924	87.4160	115.0189	204.7282
STSM_c60_12_2_Wf25_Tf60	3.9421	3.9459	3.9421	**3.9598**	**3.9598**	**3.9598**	7.3086	100.3619	113.4457	124.0396	185.6677	393.9685
STSM_c60_12_3_Wf25_Tf60	4.6486	4.7520	**4.8303**	**4.8303**	**4.8303**	**4.8303**	4.7805	88.5578	86.7858	91.4642	129.4033	373.7768
STSM_c60_12_4_Wf25_Tf60	4.0876	4.3004	**4.3275**	**4.3275**	**4.3275**	**4.3275**	6.6449	100.8419	93.8195	112.8896	171.1242	402.5906
STSM_c60_12_5_Wf25_Tf60	3.5724	3.7863	3.8017	3.8017	3.8017	**3.8039**	6.7885	112.7140	124.2569	132.9490	168.3619	337.0604
STSM_c60_12_6_Wf25_Tf60	3.1733	3.2514	3.2592	3.3049	**3.3548**	**3.3548**	5.5494	106.1913	104.2180	122.8342	136.2410	177.9632
STSM_c60_12_7_Wf25_Tf60	3.6965	3.5156	3.6965	**3.8633**	**3.8633**	**3.8633**	8.3091	124.0270	135.1714	146.9474	192.9509	415.9389
STSM_c60_12_8_Wf25_Tf60	3.5972	3.4636	**3.6759**	**3.6759**	**3.6759**	**3.6759**	5.5628	111.3648	114.9324	130.9928	169.6236	313.4555
STSM_c60_12_9_Wf25_Tf60	3.6877	**3.9640**	**3.9640**	3.9596	**3.9640**	**3.9640**	5.0215	106.3236	123.7089	123.7566	165.0307	343.3478
STSM_c60_12_10_Wf25_Tf60	3.9391	3.8737	3.9704	**3.9812**	**3.9812**	**3.9812**	6.7810	98.1170	109.5882	127.0347	182.0349	414.1833
STSM_c60_12_1_Wf12-5_Tf90	5.5066	5.2448	5.7500	5.7655	**5.7776**	**5.7776**	2.3850	65.8500	44.1981	54.3693	134.9812	385.2594
STSM_c60_12_2_Wf12-5_Tf90	5.2433	5.2524	**5.5833**	**5.5833**	**5.5833**	**5.5833**	2.6582	69.4539	57.8022	101.4496	414.6294	2069.4738
STSM_c60_12_3_Wf12-5_Tf90	5.6390	5.0976	**5.7286**	**5.7286**	**5.7286**	**5.7286**	2.4031	68.5153	55.7336	63.3229	225.7595	872.2101
STSM_c60_12_4_Wf12-5_Tf90	5.1462	5.4619	5.5669	**5.6117**	**5.6117**	**5.6117**	2.8610	57.5870	57.6327	70.9460	216.9346	1904.9969
STSM_c60_12_5_Wf12-5_Tf90	**5.7490**	5.7168	**5.7490**	**5.7490**	**5.7490**	**5.7490**	2.4456	62.5716	60.5600	70.8758	127.3747	871.9879
STSM_c60_12_6_Wf12-5_Tf90	5.5047	**5.5559**	**5.5559**	**5.5559**	**5.5559**	**5.5559**	2.8432	65.0964	60.5588	66.9923	190.8727	867.3575
STSM_c60_12_7_Wf12-5_Tf90	5.5279	5.4533	**5.7061**	**5.7061**	**5.7061**	**5.7061**	2.4781	72.9719	60.3661	83.5022	179.9716	1566.4406
STSM_c60_12_8_Wf12-5_Tf90	**5.4736**	4.9246	**5.4736**	**5.4736**	**5.4736**	**5.4736**	3.3094	78.3622	88.0674	124.4634	327.7584	1679.8830
STSM_c60_12_9_Wf12-5_Tf90	5.0745	5.2109	5.3197	5.3543	**5.3596**	**5.3596**	4.5551	74.9225	82.1746	110.3169	247.6428	2204.1870
STSM_c60_12_10_Wf12-5_Tf90	5.5781	5.2649	**5.7604**	**5.7604**	**5.7604**	**5.7604**	2.2301	58.4285	52.5357	71.4133	113.1881	566.0928
STSM_c60_12_1_Wf12-5_Tf75	4.9443	4.8513	**5.2279**	**5.2279**	**5.2279**	**5.2279**	4.2709	77.2775	78.0905	99.8830	165.3338	1685.1748
STSM_c60_12_2_Wf12-5_Tf75	4.7780	4.7968	5.0964	**5.1197**	**5.1197**	**5.1197**	3.5259	81.2134	91.3775	129.2035	324.5923	3708.0222
STSM_c60_12_3_Wf12-5_Tf75	4.9258	5.0872	**5.2435**	**5.2435**	**5.2435**	**5.2435**	4.0064	78.8456	73.1175	91.9901	182.2727	1008.1529
STSM_c60_12_4_Wf12-5_Tf75	4.9006	5.1293	4.9375	**5.1416**	5.1293	**5.1416**	4.1193	80.7601	86.6242	111.1064	257.6471	2221.3376
STSM_c60_12_5_Wf12-5_Tf75	4.3132	4.1362	4.3226	4.4760	**4.5618**	**4.5618**	5.8712	87.9523	104.4498	154.9304	382.9253	3427.8149
STSM_c60_12_6_Wf12-5_Tf75	4.2771	4.0287	**4.3481**	**4.3481**	**4.3481**	**4.3481**	5.1084	95.3078	105.7635	136.5887	230.4836	1297.6855
STSM_c60_12_7_Wf12-5_Tf75	4.6342	4.5628	4.5810	**4.7403**	**4.7403**	**4.7403**	5.7335	105.9455	112.5144	142.3870	398.4774	3768.3559
STSM_c60_12_8_Wf12-5_Tf75	4.0351	4.5276	4.5579	**4.6790**	**4.6790**	**4.6790**	6.2053	92.3490	91.8528	133.5928	254.6292	1934.4967
STSM_c60_12_9_Wf12-5_Tf75	4.3364	4.9625	**5.1997**	5.1876	**5.1997**	**5.1997**	4.9446	84.4399	62.1639	90.6096	129.2090	814.3490
STSM_c60_12_10_Wf12-5_Tf75	4.7296	4.7046	4.9183	**5.0401**	**5.0401**	**5.0401**	3.9727	80.7794	84.3630	104.8995	191.2982	2180.1435
STSM_c60_12_1_Wf12-5_Tf60	4.0790	**4.6866**	4.6136	**4.6866**	**4.6866**	**4.6866**	4.9431	81.7832	74.1336	90.1975	120.4882	475.6193
STSM_c60_12_2_Wf12-5_Tf60	3.7493	3.8748	3.9192	3.9192	**3.9325**	**3.9325**	6.3762	95.5789	103.9744	144.2867	248.6884	959.7797
STSM_c60_12_3_Wf12-5_Tf60	4.3761	**4.7754**	4.7607	**4.7754**	**4.7754**	**4.7754**	4.0898	83.4859	87.3460	101.1399	196.4607	2605.6792
STSM_c60_12_4_Wf12-5_Tf60	3.9297	4.0539	**4.2919**	**4.2919**	**4.2919**	**4.2919**	6.1308	94.6704	100.5323	137.3664	252.3394	1212.5255
STSM_c60_12_5_Wf12-5_Tf60	3.3277	3.5935	3.5458	**3.7208**	**3.7208**	**3.7208**	6.0105	106.0885	116.7447	133.8756	227.0442	1029.3909
STSM_c60_12_6_Wf12-5_Tf60	2.8374	3.0901	3.3548	3.3548	**3.3653**	**3.3653**	5.6934	98.6656	98.4402	112.0250	140.7067	316.6168
STSM_c60_12_7_Wf12-5_Tf60	3.4996	3.3783	**3.8011**	3.7021	**3.8011**	**3.8011**	7.4473	125.8111	132.1124	145.0633	226.8241	946.1685
STSM_c60_12_8_Wf12-5_Tf60	3.5021	3.4548	3.6872	3.6872	**3.6945**	**3.6945**	5.1737	106.0972	107.5046	119.5732	187.0179	686.9838
STSM_c60_12_9_Wf12-5_Tf60	3.5624	3.8829	**3.9481**	**3.9481**	**3.9481**	**3.9481**	5.4866	106.0334	115.9209	149.3930	237.4099	657.4959
STSM_c60_12_10_Wf12-5_Tf60	3.8185	3.9296	3.8856	**3.9495**	3.9393	3.9375	5.9062	109.8266	109.0889	143.4043	210.2703	1049.8073

Table 10: Detailed results for the instances with overlapping sets of possible time slots. Bold expected revenues indicate the best known solution.

	Exact Exp. Rev.			CPU time (s)		
	LSH	SAA Fixed	SAA Full	LSH	SAA Fixed	SAA Full
STSM_c60_8_1_Wf12-5o2_Tf90	3.8091	3.8091	**3.8482**	0.1122	496.7008	4803.4290
STSM_c60_8_2_Wf12-5o2_Tf90	3.5833	3.5922	**3.7271**	0.1041	1138.4635	7770.2280
STSM_c60_8_3_Wf12-5o2_Tf90	3.8752	3.8752	**3.8933**	0.0996	176.8945	1928.3496
STSM_c60_8_4_Wf12-5o2_Tf90	3.7526	3.8027	**3.8368**	0.1240	446.3710	6349.6148
STSM_c60_8_5_Wf12-5o2_Tf90	3.7394	3.7394	**3.7757**	0.1065	611.0241	8574.7235
STSM_c60_8_6_Wf12-5o2_Tf90	3.7723	3.7723	**3.7785**	0.1182	1026.2595	10678.3739
STSM_c60_8_7_Wf12-5o2_Tf90	3.7714	**3.7775**	**3.7775**	0.1125	791.9994	2793.7330
STSM_c60_8_8_Wf12-5o2_Tf90	3.7278	**3.7351**	**3.7351**	0.1239	730.1920	19331.7780
STSM_c60_8_9_Wf12-5o2_Tf90	3.5243	3.5243	**3.5984**	0.1432	1705.1862	41888.1546
STSM_c60_8_10_Wf12-5o2_Tf90	**3.6705**	**3.6705**	**3.6705**	0.1364	940.9034	4078.8893
STSM_c60_8_1_Wf12-5o2_Tf75	3.3292	3.3292	**3.4141**	0.1411	1830.8676	30803.0189
STSM_c60_8_2_Wf12-5o2_Tf75	2.9868	3.3875	**3.4222**	0.1364	772.5951	4302.2054
STSM_c60_8_3_Wf12-5o2_Tf75	3.1212	3.2828	**3.3570**	0.1581	1319.1536	82933.8089
STSM_c60_8_4_Wf12-5o2_Tf75	3.1619	**3.3958**	3.3812	0.1524	1126.0975	39800.3692
STSM_c60_8_5_Wf12-5o2_Tf75	3.2183	3.3855	**3.4278**	0.1347	1234.1769	20971.7485
STSM_c60_8_6_Wf12-5o2_Tf75	3.0200	3.2366	**3.3497**	0.1668	908.8826	23894.2788
STSM_c60_8_7_Wf12-5o2_Tf75	2.9472	**3.1960**	**3.1960**	0.1640	1827.0931	25937.2847
STSM_c60_8_8_Wf12-5o2_Tf75	2.7794	2.9490	**3.0812**	0.1763	2390.5861	54197.6124
STSM_c60_8_9_Wf12-5o2_Tf75	2.6080	3.1074	**3.2148**	0.1522	1822.7014	10039.0893
STSM_c60_8_10_Wf12-5o2_Tf75	2.9915	3.0929	**3.1054**	0.1781	1257.8017	19733.4258
STSM_c60_8_1_Wf12-5o2_Tf60	2.7330	2.8238	**2.9825**	0.1633	541.1096	9784.6687
STSM_c60_8_2_Wf12-5o2_Tf60	2.1463	2.8836	**2.9725**	0.1561	369.8185	2268.8561
STSM_c60_8_3_Wf12-5o2_Tf60	2.4849	2.8153	**2.8566**	0.1768	489.8252	4643.4262
STSM_c60_8_4_Wf12-5o2_Tf60	2.5780	2.8140	**2.9355**	0.1569	795.0578	17883.8778
STSM_c60_8_5_Wf12-5o2_Tf60	2.7772	2.8611	**2.8769**	0.1621	1011.1436	7311.9611
STSM_c60_8_6_Wf12-5o2_Tf60	2.3956	2.7231	**2.8437**	0.1881	478.5715	2665.4961
STSM_c60_8_7_Wf12-5o2_Tf60	2.1989	2.4517	**2.7145**	0.1673	926.8967	6718.1597
STSM_c60_8_8_Wf12-5o2_Tf60	2.0286	2.5787	**2.6270**	0.1786	414.6327	2377.4768
STSM_c60_8_9_Wf12-5o2_Tf60	2.1560	2.1931	**2.3875**	0.1704	169.9912	1202.7767
STSM_c60_8_10_Wf12-5o2_Tf60	2.0510	2.1366	**2.2506**	0.2248	134.6988	925.7496
STSM_c60_12_1_Wf12-5o2_Tf90	5.7974	**5.8249**		2.3256	378.4126	
STSM_c60_12_2_Wf12-5o2_Tf90	**5.7550**	**5.7550**		2.5474	1962.8252	
STSM_c60_12_3_Wf12-5o2_Tf90	**5.8939**	**5.8939**		2.0300	287.9451	
STSM_c60_12_4_Wf12-5o2_Tf90	5.7387	**5.7933**		2.3546	973.0745	
STSM_c60_12_5_Wf12-5o2_Tf90	5.8454	**5.8663**		2.3382	475.6658	
STSM_c60_12_6_Wf12-5o2_Tf90	**5.6214**	**5.6214**		2.5086	2396.2897	
STSM_c60_12_7_Wf12-5o2_Tf90	5.8165	**5.8278**		2.2851	1021.1384	
STSM_c60_12_8_Wf12-5o2_Tf90	5.3339	**5.4822**		4.0867	7250.0155	
STSM_c60_12_9_Wf12-5o2_Tf90	5.5364	**5.6307**		2.9939	3186.1611	
STSM_c60_12_10_Wf12-5o2_Tf90	5.8258	**5.8744**		2.0144	349.8826	
STSM_c60_12_1_Wf12-5o2_Tf75	5.2630	**5.3346**		4.1298	5402.3894	
STSM_c60_12_2_Wf12-5o2_Tf75	5.0978	**5.1974**		3.3740	14447.6435	
STSM_c60_12_3_Wf12-5o2_Tf75	5.0251	**5.2953**		3.4756	5928.6411	
STSM_c60_12_4_Wf12-5o2_Tf75	4.9754	**5.2042**		3.4820	7656.0720	
STSM_c60_12_5_Wf12-5o2_Tf75	4.4311	**4.6868**		7.4584	63771.7534	
STSM_c60_12_6_Wf12-5o2_Tf75	4.1540	**4.3882**		5.3126	9362.9319	
STSM_c60_12_7_Wf12-5o2_Tf75	4.7184	**4.9418**		5.8420	30246.9255	
STSM_c60_12_8_Wf12-5o2_Tf75	4.1531	**4.7199**		6.0980	10533.0046	
STSM_c60_12_9_Wf12-5o2_Tf75	4.4392	**5.2466**		5.2953	2497.1227	
STSM_c60_12_10_Wf12-5o2_Tf75	4.9871	**5.0715**		3.6438	9703.8714	
STSM_c60_12_1_Wf12-5o2_Tf60	4.3852	**4.7251**		4.0970	2209.5098	
STSM_c60_12_2_Wf12-5o2_Tf60	**3.9438**	3.9354		5.6232	10980.0181	
STSM_c60_12_3_Wf12-5o2_Tf60	4.0510	**4.8840**		5.4026	7141.3359	
STSM_c60_12_4_Wf12-5o2_Tf60	4.0184	**4.3244**		6.3066	12546.3243	
STSM_c60_12_5_Wf12-5o2_Tf60	3.5936	**3.8084**		5.3802	6986.7271	
STSM_c60_12_6_Wf12-5o2_Tf60	3.1447	**3.3653**		4.9358	603.6254	
STSM_c60_12_7_Wf12-5o2_Tf60	3.3955	**3.8703**		7.2837	7756.2249	
STSM_c60_12_8_Wf12-5o2_Tf60	3.5997	**3.7108**		5.0177	4120.8814	
STSM_c60_12_9_Wf12-5o2_Tf60	3.6352	**4.0073**		4.7563	6936.4447	
STSM_c60_12_10_Wf12-5o2_Tf60	3.8829	**3.9784**		6.5914	8440.9017	

Table 11: Detailed results of the fixed a priori route SAA method without non-anticipation and ascending time slot constrained models.

	Exact Expected Revenue												CPU time (s)											
	Fixed Route SAA without Non-Anticip.						Fixed Route SAA with Asc. Time Slots						Fixed Route SAA without Non-Anticip.						Fixed Route SAA with Asc. Time Slots					
	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$
STSM_c60_8_1_Wf12-5_Tf90	3.7271	3.7615	3.7615	3.7615	3.7615	3.7615	3.7500	3.7615	3.7615	3.7615	3.7615	3.7615	2.4183	2.5801	3.9271	7.4264	15.8959	45.0299	3.3011	3.9169	7.3390	14.6990	31.7612	376.2952
STSM_c60_8_2_Wf12-5_Tf90	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	3.4688	2.6040	3.1202	5.2243	9.0929	24.9866	83.2923	3.4324	5.3698	8.6577	20.5252	63.1485	471.0335
STSM_c60_8_3_Wf12-5_Tf90	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	3.8750	2.2741	2.5159	3.6938	6.2540	14.3819	28.8374	2.9344	4.1879	6.5308	13.3571	29.1872	112.8012
STSM_c60_8_4_Wf12-5_Tf90	3.7500	3.7526	3.7526	3.7526	3.7526	3.7526	3.7500	3.7526	3.7526	3.7526	3.7526	3.7526	2.2837	2.9607	4.4024	7.2868	13.3788	36.1131	3.3101	4.7490	8.2128	14.6286	36.3415	217.0630
STSM_c60_8_5_Wf12-5_Tf90	3.6094	3.6094	3.6094	3.6094	3.6094	3.6094	3.5535	3.6094	3.6094	3.6094	3.6094	3.6094	2.5978	2.9894	4.7340	8.1718	19.9849	64.8780	3.7408	4.7346	9.2928	16.2516	51.1763	403.5600
STSM_c60_8_6_Wf12-5_Tf90	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	3.6723	2.7666	2.7376	4.6795	8.0213	24.8834	98.6641	3.2979	4.1457	9.7408	17.0279	46.8285	464.5423
STSM_c60_8_7_Wf12-5_Tf90	3.4583	3.6309	3.6309	3.6309	3.6309	3.6309	3.6265	3.6309	3.6309	3.6309	3.6309	3.6309	2.9208	3.0603	4.9243	8.2980	19.8742	82.5935	3.4216	5.1894	8.5727	16.0881	52.7549	404.9751
STSM_c60_8_8_Wf12-5_Tf90	3.5436	3.6016	3.6016	3.6016	3.6016	3.6016	3.5436	3.5862	3.6016	3.6016	3.6016	3.6016	2.6729	3.5160	4.6316	10.1374	26.7600	107.8462	3.4643	5.8256	7.9409	18.7856	62.3044	575.8033
STSM_c60_8_9_Wf12-5_Tf90	3.3836	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.3914	3.1187	3.9878	7.0661	13.6640	58.2805	350.1663	4.5906	6.0472	11.4334	25.2074	87.4793	601.4894
STSM_c60_8_10_Wf12-5_Tf90	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	3.5177	2.9366	3.4231	5.8410	8.6144	23.7610	100.1150	3.8700	5.1000	10.1624	16.0353	68.0360	558.9897
STSM_c60_8_1_Wf12-5_Tf75	2.9796	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.2658	3.0558	3.5040	6.4973	11.0704	31.5784	107.8857	3.6149	5.6549	8.9807	16.7690	55.4372	447.9929
STSM_c60_8_2_Wf12-5_Tf75	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	3.3875	2.6311	2.9497	4.5604	8.0277	14.2662	30.0408	3.4786	4.7057	6.9731	11.7955	22.4847	62.3979
STSM_c60_8_3_Wf12-5_Tf75	3.2301	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.2552	3.1525	3.4958	6.0082	12.4979	30.9285	135.6080	3.7003	4.8488	8.7547	17.5858	62.2315	460.8348
STSM_c60_8_4_Wf12-5_Tf75	3.3945	3.3945	3.3945	3.3958	3.3945	3.3945	3.3945	3.3958	3.3958	3.3958	3.3958	3.3958	3.1897	3.9581	5.6595	9.2092	20.3343	75.7201	3.6125	5.5472	8.3418	14.2431	35.5176	297.8309
STSM_c60_8_5_Wf12-5_Tf75	3.2737	3.3157	3.3404	3.3404	3.3404	3.3404	3.2746	3.3404	3.3404	3.3404	3.3404	3.3404	3.0474	3.4143	6.0717	12.7319	31.6774	79.7117	3.6609	4.7409	9.6385	19.1618	64.2913	388.2181
STSM_c60_8_6_Wf12-5_Tf75	3.0917	3.0969	3.1224	3.1273	3.1273	3.1273	3.1224	3.1273	3.1224	3.1273	3.1273	3.1273	3.2461	4.0869	6.5142	13.8971	41.0246	266.6782	4.3119	6.2680	9.4924	18.4323	60.9058	448.0205
STSM_c60_8_7_Wf12-5_Tf75	3.1773	3.0624	3.1773	3.1773	3.1773	3.1773	3.1773	3.1773	3.1773	3.1773	3.1773	3.1773	3.1575	3.9364	5.4151	14.3135	25.4096	78.8207	3.8969	5.8379	7.4699	16.2835	33.7906	340.6008
STSM_c60_8_8_Wf12-5_Tf75	2.8682	2.8682	2.8682	2.8682	2.8682	2.8682	2.8682	2.8849	2.8849	2.8849	2.8849	2.8849	3.1596	3.8662	6.5449	16.3215	31.2751	126.8162	4.0825	5.2869	8.6509	19.8237	50.0239	506.4594
STSM_c60_8_9_Wf12-5_Tf75	3.0365	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	3.0406	2.9899	4.0061	6.4390	15.1562	36.4679	251.5266	4.0041	5.0414	8.4999	14.7995	54.2874	381.5282
STSM_c60_8_10_Wf12-5_Tf75	2.8253	2.8646	2.9374	2.9374	2.9374	2.9374	2.8646	2.9374	2.9374	2.9374	2.9374	2.9374	3.3371	4.3536	9.3516	29.0292	69.2299	341.3582	3.9499	5.6559	9.5786	17.7568	56.0491	510.8178
STSM_c60_8_1_Wf12-5_Tf60	2.7569	2.8193	2.8193	2.8193	2.8238	2.8238	2.7750	2.8238	2.8238	2.8238	2.8238	2.8238	3.2557	3.9000	5.1441	9.0675	18.9008	45.5700	3.7897	4.9560	7.6189	13.9170	27.5799	171.9146
STSM_c60_8_2_Wf12-5_Tf60	2.8836	2.8836	2.8836	2.8836	2.8836	2.8836	2.7885	2.7885	2.7885	2.7885	2.7885	2.7885	3.2223	3.8018	5.5926	9.6311	18.3875	43.3384	4.0331	5.1990	7.3350	11.4290	22.2420	46.7502
STSM_c60_8_3_Wf12-5_Tf60	2.6838	2.6838	2.6838	2.6838	2.7082	2.7082	2.6499	2.6499	2.6499	2.6499	2.6499	2.6499	3.2902	3.6742	5.7314	8.9747	23.4747	54.1036	4.1055	4.8997	7.7262	13.2604	35.4110	212.8174
STSM_c60_8_4_Wf12-5_Tf60	2.8140	2.8140	2.8140	2.8140	2.8140	2.8140	2.6835	2.7111	2.7111	2.7111	2.7111	2.7111	3.4378	4.4670	6.1328	10.5467	22.0836	58.0159	4.2688	5.6767	8.4842	13.7978	30.6732	117.3729
STSM_c60_8_5_Wf12-5_Tf60	2.7709	2.7709	2.7351	2.7709	2.7351	2.7709	2.7543	2.7709	2.7709	2.7709	2.7709	2.7709	3.8753	4.5008	7.1720	11.7259	28.5874	88.0497	4.2550	5.1070	7.1264	11.0616	21.5481	45.4145
STSM_c60_8_6_Wf12-5_Tf60	2.6385	2.6385	2.6385	2.6385	2.6385	2.6385	2.5743	2.5798	2.5798	2.5798	2.5798	2.5798	3.5764	4.3349	5.8300	10.9631	23.5014	97.9710	4.3001	5.7772	7.3826	12.0223	26.1502	150.4211
STSM_c60_8_7_Wf12-5_Tf60	2.4517	2.3650	2.4517	2.4517	2.4517	2.4517	2.3785	2.3785	2.3785	2.3785	2.3785	2.3785	3.4163	5.3658	8.3058	16.5302	34.7903	123.0234	4.5202	5.7700	8.3511	13.9507	29.0170	111.6655
STSM_c60_8_8_Wf12-5_Tf60	2.4467	2.4788	2.4788	2.4788	2.4788	2.4788	2.3983	2.3983	2.3983	2.3983	2.3983	2.3983	3.1733	3.7920	5.5009	11.9789	28.6332	146.0873	4.0354	4.9971	7.3054	12.8555	28.1082	123.4183
STSM_c60_8_9_Wf12-5_Tf60	2.1931	2.1750	2.1931	2.1931	2.1931	2.1931	2.1931	2.1931	2.1931	2.1931	2.1931	2.1931	3.4964	3.6102	4.9818	8.1620	16.4449	36.8356	4.1642	5.0193	7.1409	11.0895	21.2128	46.7235
STSM_c60_8_10_Wf12-5_Tf60	2.0822	2.1178	2.1178	2.1366	2.1366	2.1366	2.1366	2.1366	2.1366	2.1366	2.1366	2.1366	3.9116	4.0517	6.0174	8.1891	15.3016	36.3302	4.5699	5.0887	8.0286	11.8658	20.9211	45.3334
STSM_c60_12_1_Wf12-5_Tf90	5.2464	5.7500	5.7482	5.7655	5.7776		5.5363	5.7776	5.7500	5.7776	5.7776		83.7416	50.8089	61.1498	79.3689	109.8826		52.2678	44.2673	53.1667	126.0778	333.7008	
STSM_c60_12_2_Wf12-5_Tf90	5.2433	5.5756	5.5833	5.5833	5.5833		5.5756	5.5833	5.5736	5.5833	5.5833		87.1618	70.4414	74.6887	137.7360	481.0526		59.0494	60.7431	87.4320	339.3201	1374.2260	
STSM_c60_12_3_Wf12-5_Tf90	5.0976	5.7286	5.7286	5.7286	5.7286		5.6623	5.7286	5.7286	5.7286	5.7286		89.9179	64.8665	70.2441	114.4821	186.8967		54.6424	53.5981	59.7212	160.6991	808.7656	
STSM_c60_12_4_Wf12-5_Tf90	5.4619	5.6117	5.6117	5.6117	5.6117		5.5240	5.6117	5.6117	5.6117	5.6037		75.0044	78.8100	73.2261	98.7951	217.0527		57.8245	60.8042	71.1250	160.4782	1294.6037	
STSM_c60_12_5_Wf12-5_Tf90	5.6067	5.7490	5.7490	5.7490	5.7490		5.7490	5.7490	5.7490	5.7490	5.7490		85.0054	73.5252	72.6361	87.3116	234.9051		59.3452	61.5641	70.1823	112.6171	622.5531	
STSM_c60_12_6_Wf12-5_Tf90	5.5559	5.5559	5.5559	5.5559	5.5559		5.5559	5.5559	5.5559	5.5559	5.5559		81.3550	72.1905	74.7952	90.9266	154.3548		62.0900	59.1844	66.7256	163.5777	724.7494	
STSM_c60_12_7_Wf12-5_Tf90	5.4532	5.7061	5.7061	5.7061	5.7061		5.7061	5.6414	5.7061	5.7061	5.7061		91.6315	74.4922	78.8027	93.9101	270.6743		65.2318	54.7727	95.0020	148.2436	1159.7824	
STSM_c60_12_8_Wf12-5_Tf90	4.7170	5.4736	5.4736	5.4736	5.4736		5.4736	5.4736	5.4736	5.4736	5.4736		96.3188	94.7179	105.2276	122.9318	436.6353		71.4073	83.8464	95.4203	261.2101	1199.8132	
STSM_c60_12_9_Wf12-5_Tf90	5.2454	5.3596	5.3596	5.3596	5.3596		5.3121	5.3543	5.3596	5.3596	5.3596		76.7321	97.2040	101.7171	138.6826	270.1679		63.8044	77.2254	120.1937	217.1184	1719.5768	
STSM_c60_12_10_Wf12-5_Tf90	5.2732	5.7604	5.7604	5.7604	5.7604		5.7504	5.7604	5.7604	5.7604	5.7604		81.6896	68.2787	72.0228	81.5619	170.6479		53.7201	53.3864	71.4326	126.3429	526.0402	
STSM_c60_12_1_Wf12-5_Tf75	4.8513	5.2279	5.2279	5.2279	5.2279		5.2279	5.2279	5.2279	5.2279	5.2279		98.3162	98.8826	109.2436	134.7372	365.6516		78.0342	83.8173	100.5930	133.0182	1041.1875	
STSM_c60_12_2_Wf12-5_Tf75	4.9806	4.9763	5.1197	5.1197	5.1197		5.0272	5.1197	5.1197	5.1197	5.1197		112.8603	109.1091	131.4278	853.9698	1710.3347		80.6986	85.4832	105.6722	182.2824	1419.9356	
STSM_c60_12_3_Wf12-5_Tf75	5.0365	5.1533	5.2435	5.2435	5.2435		5.0386	5.1418	5.2122	5.2112	5.2122		102.8032	88.1420	82.5830	113.3189	192.1517		79.3797	72.9448	91.8609	145.6492	846.5322	
STSM_c60_12_4_Wf12-5_Tf75	5.0490	4.9538	5.1416	5.1416	5.1416		5.1416	5.1416	5.1416	5.1416	5.1416		103.6698	99.9329	115.8350	319.3993	1296.6096		82.1390	84.3757	105.2749	197.2661	1063.4402	
STSM_c60_12_5_Wf12-5_Tf75	4.2956	4.5317	4.5618	4.5618	4.5618		4.5618	4.4836	4.5618	4.5618	4.5618		124.1180	121.9281	171.3355	735.2106	3637.0891		95.6363	114.5503	145.2865	218.8141	1717.4212	
STSM_c60_12_6_Wf12-5_Tf75	4.2632	4.3481	4.3481	4.3481	4.3481		4.3481	4.3481	4.3481	4.3481	4.3481		119.0631	127.8721	166.9888	450.8146	1770.3602		97.8766	102.5467	125.3987	195.9718	901.4580	
STSM_c60_12_7_Wf12-5_Tf75	4.6020	4.6710	4.7155	4.7403	4.7403		4.7048	4.7403	4.7403	4.7403	4.7403		149.2846	146.1491	186.3191	1411.9680	4833.8691		105.8977	106.3960	126.0700	222.7280	2071.7740	
STSM_c60_12_8_Wf12-5_Tf75	4.4945	4.6790	4.6790	4.6790	4.6790		4.6790	4.6790	4.6790	4.6790	4.6790		105.6408	114.5681	147.5254	434.4093	1342.2964		94.6567	97.4407	121.5170	186.8270	1113.3455	
STSM_c60_12_9_Wf12-5_Tf75	5.1997	5.1997	5.1997	5.1997	5.1997		5.1997	5.1997	5.1997	5.1997	5.1997		81.5818	75.1289	82.4635	97.3169	187.5717		71.2577	61.0995	85.0182	124.2602	701.4577	
STSM_c60_12_10_Wf12-5_Tf75	4.8898	5.0401	5.0401	5.0401	5.0401		5.0355	5.0401	5.0401	5.0401	5.0401		102.0338	104.7939	115.7785	173.3158	1027.9595		83.1579	84.9354	100.3256	164.4683	1228.4116	
STSM_c60_12_1_Wf12-5_Tf60	4.6866	4.6866	4.6866	4.6866	4.6866		4.6866	4.6866	4.6866	4.6866	4.6866		109.4825	92.9320	100.4600	121.2256	206.6856		78.0694	74.0172	84.3795	95.5767	328.2925	
STSM_c60_12_2_Wf12-5_Tf60	3.8828	3.8962	3.9192	3.9192	3.9271		3.8610	3.8928	3.8928	3.8928	3.8928		116.9711	135.3956	180.1247	431.2156	2385.3898		116.6058	122.7827	134.8512	158.9862	388.5506	
STSM_c60_12_3_Wf12-5_Tf60	4.7754	4.7754	4.7754	4.7754	4.7754		4.6140	4.5733	4.6140	4.6140	4.6140		101.6079	104.6101	118.9192	212.8830	1097.4690		94.8699	99.7918	107.5646	144.0709	818.6896	
STSM_c60_12_4_Wf12-5_Tf60	4.0332	4.2919	4.2919	4.2919	4.2919		4.1793	4.2919	4.2919	4.2919	4.2919		117.0939	114.5617	150.8606	427.9423	1535.0184		91.6307	89.6395	113.2007	151.3897	437.6976	
STSM_c60_12_5_Wf12-5_Tf60	3.6824	3.7208	3.7208	3.7208	3.7208		3.6824	3.5831	3.6824	3.6824	3.6824		129.5768	134.5546	188.1788	539.3760	6775.7358		113.6634	116.5301	124.3152	139.7335	374.2221	
STSM_c60_12_6_Wf12-5_Tf60	3.0322	3.2697	3.3548	3.3548	3.3548		3.2697	3.2697	3.2727	3.2727	3.2727		134.0134	128.2238	137.4901	172.6271	353.4880		106.5911	112.0920	124.2948	142.5646	239.6341	
STSM_c60_12_7_Wf12-5_Tf60	3.5672	3.6668	3.6600	3.8011	3.8011		3.6712	3.6595	3.6595	3.6712	3.6712		135.8225	147.0092	183.0306	701.9748	4389.4471		127.3216	130.3706	142.0604	177.5961	402.6723	
STSM_c60_12_8_Wf12-5_Tf60	3.4862	3.5914	3.6872	3.6872	3.6872		3.5756	3.5803	3.5803	3.5756	3.5803		137.0103	141.1547	183.9404	448.9265	1488.3729		109.5195	109.5975	114.5667	147.2923	367.7725	
STSM_c60_12_9_Wf12-5_Tf60	3.9366	3.9380	3.9481	3.9481	3.9481		3.8829	3.9380	3.9481	3.9481	3.9481		141.5846	156.1215	214.8520	714.2293	1314.4637		131.4733	116.5352	136.7639	157.3700	329.9344	
STSM_c60_12_10_Wf12-5_Tf60	3.8139	3.9176	3.9495	3.9495	3.9495		3.9296	3.9296	3.9393	3.9393	3.9393		137.4779	151.0314	188.0568	382.2591	1864.6716		114.4817	117.8070	130.0936	156.9745	511.5410	