Stochastic Programming Models for a Fleet Sizing and Appointment Scheduling Problem with Random Service and Travel Times

Shutian Li\(^1\), Karmel S. Shehadeh\(^1\)*, Man Yiu Tsang\(^1\)

\(^1\)Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Abstract

We propose a new stochastic mixed-integer linear programming model for a home service fleet sizing and appointment scheduling problem (HFASP) with random service and travel times. Specifically, given a set of providers and a set of geographically distributed customers within a service region, our model solves the following problems simultaneously: (i) a fleet sizing problem that determines the number of providers required to serve customers; (ii) an assignment problem that assigns customers to providers; and (iii) a sequencing and scheduling problem that decides the sequence of appointment times of customers assigned to each provider. The objective is to minimize the fixed cost of hiring providers plus the expectation of a weighted sum of customers’ waiting time and providers’ travel time, overtime, and idle time. We compare our proposed model with an extension of an existing model for a closely related problem in the literature, theoretically and empirically. Specifically, we show that our newly proposed model is smaller and prove that it provides a tighter linear programming relaxation. Furthermore, to handle large instances observed in other application domains, we propose two optimization-based heuristics that decompose the HFASP decision process into two steps. The first step involves determining fleet sizing and assignment decisions, and the second constructs a routing plan and a schedule for each provider. We present extensive computational results to show the size and characteristics of HFASP instances that can be solved with our proposed model, demonstrating its computational efficiency over the extension. Results also show that the proposed heuristics can quickly produce high-quality solutions to large instances with an optimality gap not exceeding 5% on tested instances. Finally, we use a case study based on a service region in Lehigh County to derive insights into the HFASP.

Keywords: Fleet sizing; scheduling; routing; home services; stochastic programming

*Corresponding author.

Email addresses: shl919@lehigh.edu (Shutian Li), kas720@lehigh.edu (Karmel S. Shehadeh), mat420@lehigh.edu (Man Yiu Tsang)
1. Introduction

Home service agencies provide a wide range of services to customers at their homes, including health care, beauty treatments, fitness training, housekeeping, appliance repair service, and babysitting. The increasingly aging populations, busy lifestyles, extended work hours, and the outspread of infectious and chronic diseases have led to a substantial increase in demand for home services, especially home health care (Fikar and Hirsch, 2017; Alkaabneh et al., 2023). The cost of various home services has also been increasing in recent years. For example, 88% of home service companies raised their service prices in the past two years. The United States home service market is projected to reach $1219 Billion by 2026. With competitive pressure from the market, home service providers face the challenge of improving service quality and reducing operational costs. This motivates the need for computationally efficient optimization models for home service planning and scheduling.

Like traditional delivery services, home services require a professional provider (or service team) to travel to deliver services to geographically distributed customers. In contrast to most delivery services, however, customers must present to receive their service (Zhan and Wan, 2018). In addition, home service operators assign a provider and quote an appointment time (i.e., planned service start time) to each customer in advance to avoid delivery failure. Then, on the day of service, each provider visits customers assigned to them one by one.

Each provider must start from the main office (i.e., depot) and visit each customer in their schedule one by one before returning to the main office. If a provider arrives at a customer’s location before the scheduled service start time, the provider must wait (i.e., remains idle) until the scheduled service start time. Conversely, the customer must wait if the provider arrives after the scheduled appointment time. Moreover, each provider has a fixed service hour beyond which they experience overtime. Given the uncertainty of service time and travel time between customers, the goal is to minimize the fixed cost of establishing the providers’ fleet (i.e., provider hiring or labor cost) and the expectation of a weighted sum of customers’ waiting time and providers’ idle time, overtime, and travel time.

The HFASP is a challenging stochastic optimization problem for various reasons, foremost of which are the following. First, suppose we fix the number of providers and customers’ appointment times. In this case, the HFASP becomes similar to the multiple vehicle routing problem with time constraints and stochastic travel time, which is a challenging optimization problem (Cook, 2011; Toth and Vigo, 2014). Second, suppose we fix the number of providers and customer assignments. In this case, the HFASP reduces to a multi-server sequencing and scheduling problem with stochastic service time, which is another well-known complex optimization problem (Denton et al., 2007).

To address service time uncertainty, Zhan and Wan (2018) proposed the first two-stage stochastic mixed-integer program (SMIP) for a closely related problem to the HFASP. In the first stage, the model decides the number of providers and their routing and scheduling decisions using tradi-
tional routing variables and constraints. Then, in the second stage, the model computes providers' overtime and customers' waiting time. The objective of this model is to minimize the fixed cost of hiring providers and their total travel time plus a weighted sum of provider overtime and customer waiting time. Note that Zhan and Wan (2018) assumed that the travel time is deterministic and ignored providers' idle time. Ignoring random travel time may lead to sub-optimal solutions with excessive customers' waiting time and providers' overtime, consequently impacting service quality. Ignoring idle time may yield the underutilization of providers' time. Finally, Zhan and Wan (2018)'s results indicate that their model is challenging to solve.

In this paper, we propose a new two-stage SMIP for the HFASP, denoted as model (S). In contrast to Zhan and Wan (2018), our model incorporates both random travel and service times. In addition, our second stage includes variables and constraints to compute providers' idle time. Furthermore, instead of using traditional routing variables and constraints, we use sequencing variables and constraints to determine the order of customers assigned to each provider (equivalently, the provider’s route). We also derive an extension of Zhan and Wan (2018)'s model (denoted as model (Z)) that incorporates both random travel and service times and providers' idle time. We rigorously analyze the relative strengths and weaknesses of the two proposed models, theoretically and empirically. Specifically, we show that our newly proposed model (S) is smaller, prove that it provides a tighter linear programming relaxation, and show that it is more computationally efficient. Furthermore, to handle large instances observed in other application domains, we propose two optimization-based heuristics that decompose the HFASP decision process into two steps. The first involves determining fleet sizing and assignment decisions, and the second constructs a routing plan and a schedule for each provider.

Finally, we conduct extensive computational experiments to show the size and characteristics of problem instances that can be solved with our proposed model (S), demonstrating the significant computational performance improvements that can be gained with model (S). Specifically, our results show that model (S) can solve larger instances faster and within a reasonable time than model (Z). In addition, our results show that the proposed heuristics can quickly produce high-quality solutions to large instances with an optimality gap not exceeding 5% on tested instances. Finally, we use a case study based on a service region in Lehigh County, Pennsylvania, to derive insights into the HFASP.

1.1. Structure of the paper

The remainder of this paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we describe our problem setting. In Section 4, we introduce our proposed model (S) and model (Z) for the HFASP. In Section 5, we provide theoretical analyses of the proposed models. In Section 7, we present computational results. Finally, we draw conclusions in Section 8.
2. Relevant Literature

Planning and scheduling problems in home services have received significant attention. We refer to Fikar and Hirsch (2017); Cissé et al. (2017); Grieco et al. (2020); Di Mascolo et al. (2021) for comprehensive surveys on home service problems and applications. Next, we discuss relevant studies to our work.

Determining the number of providers (fleet size) is critical because it is a major fixed investment for home service companies. An inappropriate fleet size may lead to poor operational performance (e.g., long travel time and excessive delays and overtime) and, consequently, poor service quality. Note that the fleet sizing problem depends on the service team’s operational performance, and the latter depends on the routing and scheduling decisions. Thus, it is important to integrate fleet sizing, assignment, scheduling, and routing problems. However, most of the existing approaches for home service planning focus on a subset of these problems. For example, Restrepo et al. (2020) focus on staff dimensioning and scheduling in home care services. They take into account the uncertainty of patients’ demands. Other studies proposed models for integrated routing and scheduling (see, e.g., Han et al., 2017; Zhan et al., 2021; Cinar et al., 2021; Liu et al., 2019 and reference therein). Next, we limit the scope of this review to recent literature that is most relevant to our work. Namely, we focus on studies that proposed stochastic programming models for fleet sizing, assignment, routing, and scheduling problems arising from the home service practice.

Zhan and Wan (2018) propose the first two-stage SMIP for vehicle routing and appointment scheduling with team assignment for home services. In the first stage, the model determines the number of providers to use/hire and the route and schedule for each. The model computes teams’ overtime and customers’ waiting time in the second stage. Zhan and Wan (2018) incorporate service time uncertainty in the model, but they assume that the travel time is deterministic. The objective is to minimize the fixed hiring cost and the total deterministic travel time plus the expected overtime and waiting time. Observing the challenges of solving small instances, they propose a modified parallel saving algorithm to obtain feasible routes and use a tabu-search method to obtain near-optimal solutions. Later, Zhan et al. (2021) propose a two-stage SMIP for a single provider home service routing and appointment scheduling with stochastic service time. The objective is to minimize the providers’ travel costs and the expected second-stage cost, including providers’ idle time and customers’ waiting time. They exploit the structural properties of the proposed model and develop an L-shaped method to solve problem instances with six, eight, and ten customers.

Note that ignoring travel time uncertainty and idling cost as in Zhan and Wan (2018) and Zhan et al. (2021) may lead to sub-optimal solutions with, for example, excessive idle time, overtime, and travel time. Recent studies that incorporate stochastic travel time include Shi et al. (2018) and Hashemi Doulabi et al. (2020). Shi et al. (2018) propose a Stochastic Programming (SP) model for a home healthcare routing and scheduling problem with random service and travel times. The
objective is to minimize the hiring cost plus the expected weighted sum of transportation costs, providers’ overtime, and customers’ waiting costs. They integrate simulation and the simulated annealing algorithm to solve the model. Hashemi Doulabi et al. (2020) study a vehicle routing problem (VRP) with synchronized visits and propose a two-stage SMIP. They discuss the application of the model in home healthcare scheduling. Given the challenges of solving the proposed model with commercial solvers, they employ the L-shaped algorithm and implement a branch-and-cut method to solve the problem. Moreover, they propose valid inequalities to speed up the convergence. Yu et al. (2021) model a combination of vehicle routing with pick-up and delivery and appointment scheduling as a scenario-based mixed-integer program. They apply the model in the context of medical service routing and scheduling. Yu et al. (2021)’s model aims to minimize the operational cost plus the expected penalty cost of the early/late arrival and extra working duration of vehicles. To solve large instances, they use K-means algorithms to cluster customers into \( k \) groups and then make routing and scheduling decisions for each identified group (cluster) of customers.

As mentioned earlier, fixing customers’ assignments to providers reduces our problem to a sequencing and scheduling (SAS) problem with stochastic service time. Berg et al. (2014), Mancilla and Storer (2012), and Shehadeh et al. (2019) propose SMIP models for the single-server SAS. The SAS problem with multiple servers has been extensively studied in the healthcare scheduling literature (Gupta and Denton, 2008; Ahmadi-Javid et al., 2017). The HFASP problem is also related to VRP. We refer to Laporte et al. (1992); Kenyon and Morton (2003) for detailed discussions on formulations and methodologies for various VRP problems under random service and travel times.

Our HFASP has similar characteristics to those addressed in Zhan and Wan (2018) and Zhan et al. (2021). Nevertheless, our HFASP model is different from Zhan and Wan (2018) and Zhan et al. (2021) in the following aspects. First, we consider multiple service teams while Zhan et al. (2021)’s model focus on the routing and scheduling of one team. Second, we consider both random travel and service times and aim to minimize expected total travel time, while Zhan and Wan (2018) and Zhan et al. (2021) assume that travel time is deterministic. Various vehicle routing studies have motivated the need for hedging against travel time uncertainty to obtain high-quality routing decisions (Anderluh et al., 2020; Lecluyse et al., 2009). Third, Zhan and Wan (2018)’s second stage does not include idle time variables or objective, and Zhan et al. (2021)’s second stage does not include overtime variables or objective. Our second stage objective includes overtime, idle time, waiting time, and travel time. Fourth, recognizing that customers’ sequence is equivalent to the provider’s route, we use sequencing variables and constraints instead of routing variables and constraints to determine routing decisions. In Section 5, we show that our sequencing-based SMIP is smaller and provides a tighter LP relaxation than an extension of Zhan and Wan (2018) for the HFASP. Moreover, our model can efficiently solve realistic (previously unsolved) HFASP instances (see Section 7.2). Finally, we propose two efficient heuristics that leverage variants of our proposed model and show that they could quickly obtain near-optimal solutions to large instances.
3. Problem Setting

We start by introducing our problem settings. We consider a set of customers $P$ and a set of providers $K$. Each customer $p \in P$ must be served by exactly one provider. On the other hand, each provider has fixed service hours $[0, L]$, which are long enough to serve multiple customers. The cost of hiring one provider is $\lambda f$. Each hired provider must start from an origin (e.g., the provider’s office) and visit each customer on his/her schedule exactly once before returning to the origin. The service time $d_p$ of each customer $p \in P$ and the travel time $t_{p,p'}$ between customers $(p, p') \in P \times P$ are random with known probability distributions. Given sets $P$ and $K$, we aim to solve the following decision problems simultaneously: (1) a fleet sizing problem that determines the number of providers required to serve customers; (2) an assignment problem that determines the assignment of customers to providers; and (3) a sequencing (routing) and scheduling problem that determines the order and appointment times of customers assigned to each provider. The objective is to minimize the fixed hiring cost plus the expected operational costs associated with customers’ waiting time and providers’ idle time, overtime, and travel time.

This problem can be formulated as a two-stage SMIP. The first stage contains binary (for assignment and sequencing) and continuous (for scheduling) decision variables. Given the sequence of appointment times decided in the first stage, the second stage problem contains continuous decision variables representing what happens for each realization of service and travel times (i.e., compute waiting time, idle time, travel time, and overtime). To incorporate service and travel time uncertainty into the model, we use a Sample Average Approximation (SAA) approach. That is, we generate a sample of $N$ scenarios (each scenario consists of a vector of realizations of service and travel times which are drawn independently from the distributions corresponding to each customer and pair of customers, respectively), and then optimize the sample average of the objective. We refer to Kim et al. (2015); Kleywegt et al. (2002); Homem-de Mello and Bayraksan (2014) for the technical details and discussions on SAA.

To balance workload and better utilize providers’ time, a company may require each provider to serve a specific number of customers. Accordingly, we consider two types of service providers, namely fully used and partially used providers. Fully used providers must serve a particular number of customers (e.g., $|I| = 6$ customers), which effectively means that each provider’s sequence consists of $|I|$ customers. Accordingly, we can compute the number of providers needed to serve all customers as $|P|/|I|$. In this case, the problem is reduced to an assignment, sequencing, and scheduling problem with multiple providers. On the other hand, partially used providers can serve any number of customers but at least one customer (or any other threshold specified by the decision maker). In this case, the model will determine the number of providers to hire and the assignment, sequencing, and scheduling decisions. In Section 4, we present SMIP models for each provider type.
Table 1: Notation

<table>
<thead>
<tr>
<th>Index sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>the set of customers</td>
</tr>
<tr>
<td>$K$</td>
<td>the set of providers</td>
</tr>
<tr>
<td>$I$</td>
<td>positions in the serving sequence</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>the number of scenarios</td>
</tr>
<tr>
<td>$\lambda_f$</td>
<td>fixed cost of hiring one provider</td>
</tr>
<tr>
<td>$\lambda_o/\lambda_g/\lambda_w$</td>
<td>unit overtime/idle time/waiting time cost</td>
</tr>
<tr>
<td>$\lambda_t$</td>
<td>unit travel time cost</td>
</tr>
<tr>
<td>$L$</td>
<td>working hours</td>
</tr>
<tr>
<td>$t_{ij}^n$</td>
<td>travel time from node $i$ to node $j$ under scenario $n$</td>
</tr>
<tr>
<td>$d_i^p$</td>
<td>customer $i$’s service time under scenario $n$</td>
</tr>
</tbody>
</table>

**Deterministic decision variables**

- $x_{i,p,k}$ equals 1 if patient $p$ is assigned to the $i$th position of provider $k$’s sequence
- $a_{i,k}$ scheduled time of $i$th appointment served by provider $k$
- $z_{i,p,p',k}$ equals 1 if the customer $p$ follows the customer $p'$ on provider $k$’s serving sequence

**Random (scenario-Based) variables**

- $s_{i,k}^n$ actual start time of $i$th appointment served by provider $k$ under scenario $n$
- $g_{i,k}^n$ idle time before $i$th appointment served by provider $k$ under scenario $n$
- $o_k^n$ overtime of provider $k$ under scenario $n$

4. SMIP Models for the HFASP

In this section, we propose two SMIP formulations for the HFASP. In Section 4.1, we present our new SMIP model denoted as model (S). In Section 4.2, we derive an extension of Zhan and Wan (2018)’s model denoted as model (Z).

4.1. Model (S)

In this section, we present our proposed model, denoted as model (S). Let us first introduce the variables and parameters defining this model. For all $i \in I, p \in P, k \in K$, we define binary decision variables $x_{i,p,k}$ that equal 1 if customer $p$ is assigned to the $i$th position in the sequence of customers assigned to provider $k$. For all $i \in I, k \in K$, we define non-negative continuous decision variables $a_{i,k}$ to represent the scheduled appointment time of the $i$th customer in the schedule. We define the following scenario-based decision variables to compute waiting, idle time, and travel time in each scenario $n \in [N]$ of service and travel times. We define a non-negative continuous decision variable $s_{i,k}^n$ to represent the actual service start time of the $i$th customer in provider $k$’s schedule. We define non-negative continuous decision variable $g_{i,k}^n$ to represent provider $k$’s idle time before the $i$th customer/appointment. We define non-negative continuous decision variable $o_k^n$ to represent provider $k$’s overtime. Finally, we let non-negative parameters $\lambda_w, \lambda_o, \lambda_g,$ and $\lambda_t$ represent unit penalty cost of waiting, idle time, overtime, and travel time, respectively. A complete list of the parameters and decision variables can be found in Table 1. Using this notation, our SAA model can be stated as follows (see Appendix A for details on the derivation of this model):

$$\min_{x,z,a,s,g,o} \sum_{p \in P} \sum_{k \in K} \sum_{n \in [N]} \lambda_f x_{1,p,k} + \sum_{n \in [N]} \frac{1}{N} \left\{ \lambda_t \left[ \sum_{k \in K} \sum_{(p,p') \in P \times P} \sum_{i \in I} t_{p,p'}^n z_{i,p,p',k} + \sum_{p \in P} \sum_{p \neq p'} t_{0,p}^n x_{1,p,k} + \sum_{p \in P} t_{p,0}^n x_{0,p,k} \right] \right\}$$
\[
+ \sum_{k \in K} \sum_{i \in I} \left( \lambda^w (s^n_{i,k} - a_{i,k}) + \lambda^g g^n_{i,k} \right) + \sum_{k \in K} \lambda^o o^n_k \right) \tag{1a}
\]

s.t. \[
\sum_{i \in I} \sum_{k \in K} x_{i,p,k} = 1, \quad \forall p \in P, \tag{1b}
\]

\[
\sum_{p \in P} x_{i,p,k} \leq 1, \quad \forall i \in I, k \in K, \tag{1c}
\]

\[
x_{0,p,k} \geq x_{i,p,k} - \sum_{p' \in P; p' \neq p} x_{i+1,p',k}, \quad \forall i \in I, p \in P, k \in K, \tag{1d}
\]

\[
\sum_{p \in P} x_{i,p,k} \geq \sum_{p \in P} x_{i+1,p,k}, \quad \forall i \in [1, |I| - 1], k \in K, \tag{1e}
\]

\[
z_{i,p,p',k} \leq x_{i-1,p,k}, \quad \forall i \in [2, |I|], (p, p') \in P \times P : p \neq p', k \in K, \tag{1f}
\]

\[
z_{i,p,p',k} \leq x_{i,p',k}, \quad \forall i \in [2, |I|], (p, p') \in P \times P : p \neq p', k \in K, \tag{1g}
\]

\[
z_{i,p,p',k} \geq x_{i-1,p,k} + x_{i,p',k} - 1, \quad \forall i \in [2, |I|], (p, p') \in P \times P : p \neq p', k \in K, \tag{1h}
\]

\[
a_{i,k} \leq L \sum_{p \in P} x_{i,p,k}, \quad \forall k \in K, i \in I \tag{1i}
\]

\[
s^n_{1,k} \geq a_{i,k}, \quad \forall i \in I, k \in K, n \in [N], \tag{1j}
\]

\[
s^n_{1,k} \geq \sum_{p \in P} t^n_{0,p,x_{1,p,k}}, \quad \forall k \in K, n \in [N], \tag{1k}
\]

\[
s^n_{i,k} \geq s^n_{i-1,k} + \sum_{p \in P} d^n_p x_{i-1,p,k} + \sum_{(p,p') \in P \times P: p \neq p'} t^n_{p,p'} z_{i,p,p',k} - M_i \left(1 - \sum_{p \in P} x_{i,p,k}\right), \forall i \in [2, |I|], k \in K, n, \tag{1l}
\]

\[
g^n_{1,k} \geq s^n_{1,k} - \left(\sum_{p \in P} t^n_{0,p,x_{1,p,k}}\right), \quad \forall k \in K, n \in [N], \tag{1m}
\]

\[
g^n_{i,k} \geq s^n_{i,k} - \left(s^n_{i-1,k} + \sum_{(p,p') \in P \times P: p \neq p'} t^n_{p,p'} z_{i,p,p',k} + \sum_{p \in P} d^n_p x_{i-1,p,k}\right), \forall i \in [2, |I|], k \in K, n, \tag{1n}
\]

\[
o^n_k \geq \left(s^n_{i,k} + \sum_{p \in P} (d^n_p + t^n_{p,0}) x_{0,p,k}\right) - L, \quad \forall i \in I, k \in K, n \in [N], \tag{1o}
\]

\[(a, s, g, o, z) \geq 0, \quad x \in \{0, 1\}^{(|I|+2) \times |P| \times |K|}. \tag{1p}\]

Formulation (1) finds optimal sizing, routing, and scheduling decisions that minimize the fixed cost related to establishing the providers fleet or hiring costs (first term), and the sample average of the random operational costs consisting of total waiting time, and providers’ idle time, overtime, and travel time. Constraints (1b) ensure that each customer is assigned to exactly one position in the schedule of one provider. Constraints (1c) ensure that at most one customer is assigned to each position in provider \(k\)’s sequence. Constraints (1d) define the variable \(x_{0,p,k}\), which is equal to 1 if customer \(p\) is the last customer in the schedule of provider \(k\), and is 0 otherwise. Constraints (1e) prohibit assigning customers to position \((i + 1)\) when position \(i\) is vacant. Constraints (1f)-(1h)
Proposition 1. Let \( \tilde{d} = \max_{n \in [N], p \in P} \{d_p^n\}, \tilde{t} = \max_{n \in [N], p, p' \in P} \{t_{p,p'}^n\} \) and \( t_{1}^{\max} = \max_{n \in [N], p \in P} \{t_{0,p}^n\} \). Suppose \( (\lambda^f, \lambda^l, \lambda^w, \lambda^g, \lambda^o) > 0 \). Then, \( M_i \geq L + t_{1}^{\max} + (i-1)(\tilde{d} + \tilde{t}), \) for \( i \in [2, |I_k|] \) are valid lower
bound values for the $M_i$ constants in (11).

4.2. Model (Z)

Let us now introduce our extension of Zhan and Wan (2018)’s formulation for the HFASP, denoted as model (Z). First, we note that Zhan and Wan (2018) treats customers as nodes with customer 0 representing the depot (i.e., the set of customers is $P \cup \{0\}$) and employs routing variables and constraints to find providers’ routes. Thus, as in Zhan and Wan (2018), we define a binary decision variable $z_{p,q,k}$ that equals one if provider $k$ travels from customer $p$ to $q$, and zero otherwise, for all $p \in P \cup \{0\}$, $q \in P \cup \{0\}$, and $k \in K$. For all $p \in P$, we define a non-negative continuous decision variable $A_p$ to represent the scheduled appointment time of customer $p$. For each $p \in P \cup \{0\}$ and $n \in [N]$, we define non-negative continuous variables $S^n_p$ and $W^n_p$ to respectively represent the actual start time and waiting time of customer $p$ under scenario $n$. Finally, we define non-negative continuous decision variable $O^n_k$ to represent provider $k$’s overtime, and non-negative continuous decision variable $G^n_p$ to represent provider idle time after serving customer $p$ under scenario $n$.

Using this notation, model (Z) can be stated as follows:

$$\min z, A, W, O, G \sum_{k \in K} \sum_{p \in P} \lambda^f z_{0,p,k} + \frac{1}{N} \lambda^f \sum_{k \in K} \sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} t^p_{p,q} z_{p,q,k} + \lambda^w \sum_{p \in P} W^n_p + \lambda^o \sum_{k \in K} O^n_k \lambda^g \sum_{p \in P} G^n_p,$$

s.t. \begin{align*}
    \sum_{k \in K} \sum_{q \in P \cup \{0\}} z_{p,q,k} &= 1, \quad \forall p \in P, \quad (3a) \\
    \sum_{p \in P \cup \{0\}} z_{p,q,k} - \sum_{p \in P \cup \{0\}} z_{q,p,k} &= 0, \quad \forall q \in P, k \in K, \quad (3b) \\
    \sum_{p \in P \cup \{0\}} z_{p,0,k} &= 1, \quad \forall q \in P, k \in K, \quad (3c) \\
    \sum_{p \in P \cup \{0\}} z_{0,p,k} &= 1, \quad \forall k \in K, \quad (3d) \\
    \sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} z_{p,q,k} &\leq |I| + 1 \quad \forall k \in K, \quad (3e) \\
    A_p &\leq L \quad \forall p \in P, \quad (3f) \\
    \sum_{p \in P} \sum_{q \in P \cup \{0\}} z_{p,q,k} &\leq |P'| - 1, \quad \forall P' \subset P, k \in K, \quad (3g) \\
    S^n_0 &= 0, \quad \forall n \in [N], \quad (3h) \\
    S^n_p &\geq A_p, \quad \forall p \in P, n \in [N], \quad (3i) \\
    S^n_q &\geq S^n_p + d^n_p + t^n_{p,q} - M \left(1 - \sum_{k \in K} z_{p,q,k}\right), \quad p \in P \cup \{0\}, q \in P, n \in [N], \quad (3j) \\
    W^n_p &= S^n_p - A_p, \quad \forall p \in P, n \in [N], \quad (3k)
\end{align*}
\[ O^n_k \geq S^n_p + d^n_p + t^n_{p,0} - L - M(1 - z_{p,0,k}), \quad \forall p \in P, k \in K, n \in [N], \quad (3m) \]

\[ G^n_q \geq S^n_q - S^n_p - d^n_p - t^n_{p,q} - M \left( 1 - \sum_{k \in K} z_{p,q,k} \right), \quad \forall p \in P \cup \{0\}, q \in P, n \in [N], \quad (3n) \]

\[ (A, S, W, O, G) \geq 0, \quad z \in \{0, 1\}^{(|P|+1) \times (|P|+1) \times |K|.} \quad (3o) \]

Formulation (3) finds optimal fleet sizing, routing, and scheduling decisions that minimize the fixed cost and the sample average of the random operational cost. Constraints (3b) ensure that every customer must be visited exactly once. Constraints (3c) ensure the conservation of flow for each provider \( k \) at each customer and the depot (i.e., node 0). Constraints (3d)-(3e) ensure that every provider \( k \) needs to start from and end at the depot (i.e., node 0). Constraints (3f) ensure that the number of customers assigned to each provider is at most \( |I| \). Constraints (3g) ensure that all appointments are scheduled within service hours. Constraints (3h) are subtour elimination constraints. Note that although these constraints are not necessary for finding the optimal tour, as reported by Zhan and Wan (2018), these constraints could improve the model’s computational performance. Constraints (3i) set the actual service start time of the service team’s office (or the service start time of the workday) to 0.

For each scenario \( n \), constraints (3j)-(3k) require that the actual start time, \( S^n_q \), of each customer \( q \) to be no smaller than the scheduled start time \( A_q \), and the completion time of the preceding customer \( p \) plus the travel time from customer \( p \) to \( q \). Note that, for a sufficiently large \( M \) constant, constraints (3k) are relaxed if customer \( p \) is not followed by customer \( q \) in provider \( k \)’s schedule. Constraints (3l) compute the waiting time of each customer in each scenario as the difference between the actual start and scheduled times. Constraints (3m) and (3n) compute the overtime and idle time of each provider in each scenario. Formulation (3) extends that of Zhan and Wan (2018) in the following aspects. First, it incorporates uncertainty of travel time, which was ignored in Zhan and Wan (2018). Second, we modify Zhan and Wan (2018)’s first stage by requiring all appointments to be scheduled within the service hours. Third, we generalize Zhan and Wan (2018)’s second stage by (a) including variables and constraints to compute the random idle time; (b) including random idle time in the objective; and (c) including the random total travel time in the objective of the second stage (recall that Zhan and Wan (2018) assume that the travel time is deterministic).

Note that formulation 3 is for partially used providers (i.e., it does not require hired providers to serve a particular number of customers). However, recall that for the case of fully used providers, we require each provider to serve \( |I| \) customers, and accordingly, we know that we need \( |P|/|I| \) providers to serve all customers. Therefore, to model the fully used provider case, we remove the fixed cost (first term) from model (3)’s objective and replace constraints (3f) with

\[ \sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} z_{p,q,k} = |I| + 1 \quad \forall k \in K \quad (4a) \]

In Proposition 2, we derive a tight lower bound estimation of the Big-M coefficients involved in
Table 2: Size of formulations of HFASP with |I| positions, |P| customers, |K| caregivers and N scenarios

<table>
<thead>
<tr>
<th></th>
<th>Model (Z) fully used</th>
<th>Model (S) partially used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First-stage constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Second-stage constraints</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Size of formulations of HFASP with |I| positions, |P| customers, |K| caregivers and N scenarios

<table>
<thead>
<tr>
<th></th>
<th>Model (Z) fully used</th>
<th>Model (S) partially used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Theorem 1. Suppose (λf, λt, λw, λ0, λc) > 0. Model (S) and model (Z) for fully used providers are equivalent. In particular, given an optimal solution to the model (S), we can construct a feasible solution to the model (Z) with the same objective function value and vice versa.
Finally, in Theorem 2, we show that the LPR of model (S) provides a tighter linear relaxation than the LPR of model (Z); see Appendix E for a proof. The theoretical analyses in this section suggest that the smaller and tighter model (S) has a better computational performance than model (Z). Indeed, our computational results in Section 7.2 support this conclusion.

**Theorem 2.** Suppose $(\lambda^f, \lambda^t, \lambda^w, \lambda^g, \lambda^o) > 0$. The optimal objective value of the LPR of model (S) is greater than or equal to the optimal objective value of the LPR of model (Z).

### 6. Heuristics for the HFASP

In Section 7, we show that our proposed model (S) can efficiently solve realistic (and previously unsolved) HFASP instances to optimality. However, solution times increase as the instance size increases. Therefore, in this section, we propose two heuristics (denoted as FAS-RS and FAS-R-S) that allow for obtaining near-optimal solutions of larger instances that may be observed in applications other than the HFASP within an acceptable time. These heuristics decompose the decision process into two parts. The first part involves deciding the number of providers to hire (fleet sizing) and customer assignments to hired providers. The second part involves constructing a routing plan and a schedule for each provider. Both heuristics implement an integer program denoted as FAS to determine the number of providers to hire and customer-to-provider assignments. Then, the FAS-RS heuristic employs a single-provider variant of model (S) to obtain an optimal routing plan and a schedule for each provider. In contrast, the FAS-R-S heuristic employs a modified insertion heuristic to determine a routing plan for each provider and then an LP to determine the appointment time for each customer. We discuss the details of these heuristics in the next subsections.

#### 6.1. Two-Phase Heuristic: FAS-RS

Algorithm 1 summarizes the steps of our two-phase FAS-RS heuristic. In phase 1, we solve a fleet sizing and assignment (FAS) problem that determines the number of providers to hire and customer-to-provider assignments. In the second phase, we implement a stochastic single-provider routing and scheduling (RS) model to determine an optimal routing plan and a schedule for each provider. Next, we discuss the details of the FAS model employed in phase 1. We define a binary decision variable $u_k$, which equals one if provider $k \in K$ is hired, and a binary decision variable $y_{p,k}$, which equals one if customer $p \in P$ is assigned to provider $k$. For each $p \in P$, $k \in K$ and scenario $n \in [N]$, we define parameter $\lambda_{p,k}^n = t_{0,p}^n + t_{p,k}^n - t_{0,p}^n$, where $t_{i,j}^n$ is the travel time between customers $(i,j) \in (P \cup \{0\}) \times (P \cup \{0\})$ under scenario $n \in [N]$, and $p_k^*$ is the customer with the $k$th smallest expected travel time to the depot. For example, $p_1^*$ is the customer with the shortest expected travel time to the depot. Intuitively, $\lambda_{p,k}^n$ evaluates the additional travel time if provider $k$ visits customer $p$ under scenario $n$. Using this notation, we formulate the FAS problem as follows
Algorithm 1: The FAS-RS Heuristic

**Input**: Sets of customers \( P \), providers \( K \), and scenarios \( N \)

**Output**: Subset \( \bar{K} \subseteq K \) of hired providers, a route \( \mathbf{z} \) and a schedule \( \mathbf{a} \) for each \( k \in \bar{K} \)

**Phase 1.** Solve the FAS problem in (5), record the optimal solution \( (u^*, y^*) \), and set \( \bar{K} \leftarrow \{ k \in K : u^*_k = 1 \} \) and \( P_k \leftarrow \{ p \in P : y^*_p,k = 1 \} \), for all \( k \in \bar{K} \);

**Phase 2.** For each \( k \in \bar{K} \), solve formulation (2) and return optimal \( (x^*, a^*) \)

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in K} \lambda^I u_k + \sum_{p \in P} \sum_{k \in \bar{K}} \sum_{n \in [N]} \frac{1}{N} \lambda_{p,k} y_{p,k} \\
\text{subject to:} & \quad \sum_{p \in P} y_{p,k} \leq |I_k| u_k, \quad \forall k \in K, \quad (5b) \\
& \quad \sum_{k \in \bar{K}} y_{p,k} = 1, \quad \forall p \in P, \quad (5c) \\
& \quad u_k, y_{p,k} \in \{0,1\}, \quad \forall p \in P, k \in K. \quad (5d)
\end{align*}
\]

Formulation (5) determines the optimal number of providers and customer assignments that minimize the total hiring cost and the additional travel time of adding customers into a provider’s route. Constraints (5b) ensure that each provider serves at most \( |I_k| \) customers. Constraints (5c) ensure that each customer is assigned to exactly one provider. For the fully used provider case, we replace constraints (5b) with \( \sum_{p \in P} y_{p,k} = |I_k| u_k \) to ensure that each hired provider serves exactly \( |I_k| \) customers. Let \( P_k := \{ p \in P : y^*_p,k = 1 \} \) represent the set of customers assigned to provider \( k \), where \( y^* \) is an optimal solution to (5). In phase 2, we solve a single-provider variant of model (S) in (2) for each \( k \) with \( P \) fixed to \( P_k \) and \( |K| = 1 \) to obtain an optimal schedule and a routing plan. Our results in Section 7.3 show that our FAS-RS heuristic could quickly obtain near-optimal solutions for large HFASP instances.

6.2. Three-Phase Heuristic: FAS-R-S

In this section, we present our three-phase heuristic, donated as FAS-R-S. Algorithm 2 summarizes the steps of this heuristic. Phase 1 is the same as that of FAS-RS and involves solving the FAS model in (5) to obtain a set of hired providers and customer-to-provider assignments. Then, in phase 2, we employ a modified insertion heuristic (MIH) to determine a routing plan for each provider. Finally, given the routing plan from phase 2, we solve an LP model to determine the appointment time for each customer (phase 3).

Next, we discuss the details of the MIH employed in phase 2. Let \( P_k \) represent the set of customers assigned to provider \( k \) obtained from phase 1. We define \( R_k := \{ (1,q), (2,\cdots), (|I_k|,\cdots) \} \) as the partial route of provider \( k \), where \( (i,q) \) indicate that customer \( q \in P_k \) is in position \( i \in I_k \) and \( (i,\cdots) \) indicate that position \( i \) is empty. Finally, we let \( R_k(i,p) \) represent the new route resulting from inserting customer \( p \in P_k \) into position \( i \). For example, suppose that the current route is \( R_k := \{ (1,q), (2,q'''), (3,q'''), \cdots \} \) and we want to insert customer \( p \) between the second and
Algorithm 2: The FAS-R-S Heuristic

**Input:** Sets of customers \( P \), providers \( K \), and scenarios \( [N] \)

**Output:** Subset \( \bar{K} \subseteq K \) of hired providers, a route \( (R_k) \) and a schedule \((a)\) for each \( k \in \bar{K} \)

**Phase 1.** Solve the FAS problem in (5), record the optimal solution \((u^*, y^*)\), and set \( \bar{K} \leftarrow \{ k \in K : u_k^* = 1 \} \); and \( P_k \leftarrow \{ p \in P : y_{p,k}^* = 1 \} \), for all \( k \in \bar{K} \);

**for** \( k \in \bar{K} \)** do

**Phase 2.** Implement Algorithm 3 to obtain a route \( R_k \);

**Phase 3.** Solve model (2) with \((x, z)\) fixed according to \( R_k \) and return \( a^* \);

end

---

Figure 1: An illustration of inserting a customer into a partially constructed route.

Third customer in \( R_k \), i.e., assign \( p \) to position \( i = 3 \). Then, the new route will be \( R_k(3, p) := \{(1, q), (2, q''), (3, p), (4, q'''), \ldots \} \). Note that inserting a customer in the route may increase the provider’s travel time, change customers’ start times, and potentially increase waiting time and overtime. Thus, the MIH finds a position for each customer with the lowest insertion cost; see Figure 1. Algorithm 3 summarizes the steps of our MIH heuristic, which extends the insertion heuristic for VRP (see, e.g., Campbell and Savelsbergh (2004)) to fit the HFASP. Starting with an empty route, the MIH heuristic iteratively inserts each as-of-yet unassigned customer in \( P_k \) into a position in the partially constructed route \( R_k \) that leads to the lowest insertion cost (described next). Since the number of customers and positions are finite, the algorithm terminates in a finite number of iterations. To compute the cost on route \( R_k \), we first compute the actual start time for each customer in route \( R_k \) as follows.

\[
\text{start}(1, k, n) = t^n_{0,r_1}, \quad (6a)
\]

\[
\text{start}(i, k, n) = \text{start}(i-1, k, n) + d^n_{r_{i-1}} + t^n_{r_{i-1}, r_i}, \quad \forall i \in [2, |R_k|] \quad (6b)
\]

Recall that the start time should be greater than or equal to the scheduled appointment time, and the latter should not exceed \( L \). Accordingly, we compute the appointment time \( \text{appoint}(i, k) \) as follows (we later refine the appointment time in phase 3)

\[
\text{appoint}(i, k) = \min \left\{ \min_{n \in [N]} \{ \text{start}(i, k, n) \}, L \right\}. \quad (7)
\]

Given customers’ actual service start time and appointment times, we can compute the total waiting time \( EW(R_k) \) and overtime \( EO(R_k) \) on route \( R_k \) as follows.
Algorithm 3: The Modified Insertion Heuristic (MIH)

**Input**: A set of hired providers $K$ and the set of assigned customers $P$ to each

**Output**: A routing plan $R_k$ for each provider $k \in K$

for $k \in K$
do
  Initialize an empty route $R_k = \emptyset$ and the set of unassigned customers $\bar{P}_k \leftarrow P_k$
  while $\bar{P}_k \neq \emptyset$
do
    for $i \in [1,|R_k|+1]$ do
      // Insertion step
      Construct a new route $R_k(i,p)$ by inserting $p$ into the $i$th position in $R_k$
      for $j \in [i,|R_k(i,p)|]$ do
        Compute $start(j,k,n)$, $appoint(j,k)$ using (6a)-(6b) and (7);
        Compute the insertion cost $Insert(i,p,k)$ using equation (8);
      end
      Assign $p$ to position $i^* \in \arg \min_{i \in [1,|R_k|+1]} Insert(i,p,k)$ and Update $R_k \leftarrow R_k(i^*,p)$;
    end
  end
  $\bar{P}_k \leftarrow \bar{P}_k \setminus \{p\}$;
end

$$EW(R_k) = \sum_{i=1}^{|R_k|} \max\{start(i,k,n) - appoint(i,k),0\},$$

$$EO(R_k) = \max\{\text{start}(|R_k|,k,n) + d_{|R_k|}^n + t_{|R_k|}^n - 0,0\}. $$

Finally, we define the cost of inserting customer $p \in P_k$ into the $i$th position of provider $k$’s route $Insert(i,p,k)$ as the weighted sum of extra travel cost ($\lambda_t \cdot ET(i,p,k)$), extra waiting cost ($\lambda_w \cdot EW(i,p,k)$) and extra overtime cost ($\lambda_o \cdot EO(i,p,k)$):

$$Insert(i,p,k) = \lambda_t \cdot ET(i,p,k) + \lambda_w \cdot EW(i,p,k) + \lambda_o \cdot EO(i,p,k), \quad (8)$$

where

$$ET(i,p,k) = \sum_{n \in [N]} \frac{1}{N} \left[ t_{r_{i-1}^n,p}^n + t_{p,r_k^n}^n - t_{r_{i-1}^n,r_k^n}^n \right], \quad (9a)$$

$$EW(i,p,k) = EW(R_k(i,p)) - EW(R_k), \quad EO(i,p,k) = EO(R_k(i,p)) - EO(R_k). \quad (9b)$$

Once we obtain a routing plan $R_k$ from phase 2, in phase 3, we determine the optimal appointment times for each customer on the route by solving a single-provider LP variant of model (S) in (2) with $|K| = 1$ and sequencing variables $(x,z)$ fixed according to $R_k$.  

16
7. Computational Experiments

In this section, we present computational experiments that explore the size and characteristics of the HFASP instances that can be solved using the proposed models for HFASP. In addition, we derive insights into the HFASP. In Section 7.1, we describe the set of HFASP instances that we constructed and discuss other experimental setups. In Sections 7.2.1 and 7.2.2, we analyze solution times of the proposed models for fully used and partially used providers, respectively. In Section 7.3, we analyze solution times and solution quality of the proposed heuristics. Finally, in Section 7.4, we construct HFASP instances based on Lehigh County of PA to study the sensitivity of HFASP solutions to various parameter settings and derive insights.

7.1. Description of experiments

We construct HFASP instances in part based on benchmarks and parameters settings made in recent related literature (e.g., Zhan and Wan, 2018; Hashemi Doulabi et al., 2020; Yu et al., 2021). Each instance is characterized by the number of customers $|P|$ and providers $|K|$. We consider instances with $|P| \in \{6, \ldots, 72\}$ customers and $\lceil |P|/6 \rceil + 1$ service providers (Zhan and Wan, 2018). We generate most instances using the approach proposed in Zhan and Wan (2018). Specifically, we assume that customers are located uniformly and randomly on a $50 \times 50$ square, and the service provider’s office is at the point $[0,0]$. We set the daily working hour of each provider $L$ to 8 hours. We set the cost parameters in the objective function in part based on Zhan and Wan (2018). Specifically, we set the unit overtime cost $\lambda_o$ to 1, unit idle cost $\lambda_g$ to 0. We generate the unit waiting cost $\lambda_w$ from $U[0, 1.5]$ ($U[a,b]$ is a uniform distribution over the interval $[a,b]$), and the unit travel cost $\lambda_t$ from $U[0.1, 0.5]$. We set the unit fixed cost $\lambda_f$ to 1000. As in prior literature, we use the lognormal distribution for the service time $d_n$ truncated on the interval $[10, 50]$ with mean $\mu$ and $\sigma = 0.5\mu$, where $\mu$ is generated from $U[25, 35]$. We assume that customers in the same neighborhood spend an average of $T$ minutes (e.g., $T = 20$ minutes) traveling from one place to another. The assumption is realistic and widely adopted in recent literature (see, e.g., Nikzad et al., 2021; Tsang and Shehadeh, 2022). Accordingly, we consider two types of travel times: travel time from the provider’s office to the customer and travel time between customers. The travel times between the office and customer $p$, $t_{0,p}^n$ and $t_{p,0}^n$, follow a normal distribution $N(t_p, \frac{t_p}{\sqrt{8}})$, where $t_p$ is the Euclidean distance between the customer $p$ and the office (Hashemi Doulabi et al., 2020). Consistent with the literature, we generate the travel time between customers, $t_{p,p'}^n$, from $U[15, 25]$, for all $(p,p') \in P \times P, n \in [N]$. We set $d_0^0 = 0$ and $t_{0,0}^n = 0$ in model (Z) for all scenarios $n \in [N]$ ($p = 0$ is the provider’s office (or depot)). Note that the average number of customers that an operator may visit per day is often less than 6 in home health care and banking, and less than 10 in repair services (NAHC, 2010; Yuan et al., 2015; Zhan et al., 2021). Accordingly, we consider cases where a provider can visit 6 or 8 customers.
To decide an appropriate sample size for the proposed SAA models, we employed the Monte Carlo optimization (MCO) procedure, which provides statistical lower and upper bounds on the optimal value of the HFASP based on the optimal solution to its SAA with $N$ scenarios. This, in turn, provides a statistical estimate of the approximated relative gap between the lower and upper bounds. Applying the MCO procedure with $N = 50$, the relative approximation gaps for the HFASP instances were around 1%, whereas larger $N$ resulted in longer solution times without consistent and significant improvements in the gap. Therefore, we selected $N = 50$ for our computational experiments. In Appendix F, we provide a description of the MCO procedure and a numerical example. We refer readers to Kenyon and Morton (2003), Kleywegt et al. (2002), Shapiro et al. (2021) and references therein for further details on MCO and related technical results.

In our implementations, we add symmetry-breaking constraints (I.1) to model (S) and constraints (I.2) to model (Z) to enforce that provider $k$ is hired before provider $k + 1$. We set the values of the big-M coefficients in constraints (1l) according to Proposition 1 and those in (3k),(3m) and (3n) according to Proposition 2. We implemented our proposed models in AMPL modeling language and used CPLEX (version 20.1.0.0) as the solver with default settings. We imposed a solver time limit of 3600 seconds (1 hour) for each SAA instance. We conducted all the experiments on a computer with an Intel Xeon Silver processor with 2.10 GHz CPU and 128 Gb memory.

7.2. CPU time
In this section, we compare solution times of the proposed models for the fully used (Section 7.2.1) and partially used (Section 7.2.2) provider cases.

7.2.1. Fully used provider case
Recall that when the provider must serve $|I|$ customers, we know the number of providers needed to serve all customers (i.e., $|K| = |P|/|I|$). Hence, the problem reduces to an assignment, sequencing, and scheduling problem. Therefore, we solve problem instances with $|I| = 6$ ($|I| = 8$), $|P| \in \{6,12,\ldots,60\}$ ($|P| \in \{8,16,\ldots,56\}$), and $|K| \in \{1,2,\ldots,10\}$. For each combination of $|I|$, $|P|$, and $|K|$, we generated and solved five random SAAs as described in Section 7.1.

Table 3 presents the minimum (Min), average (Avg), and maximum (Max) solution time (in seconds) solved instances using model (S) within the imposed one-hour time limit. We observe the following from this table. First, using model (S), we were able to solve all the SAAs that corresponds to problem instances with ($|I| = 6$, $|P| \leq 36$) and ($|I| = 8$, $|P| \leq 32$) within 30 seconds. Second, the average solution time of the larger instances with $|I| = 6$ ranges from 2.3 minutes ($|P| = 42$) to 24 minutes ($|P| = 60$), and with $|I| = 8$ ranges from 46 seconds ($|P| = 40$) to 30 minutes ($|P| = 56$). Finally, we were not able to solve instances with $|I| = 8$ and $|P| = 64$ and 72 customers within the imposed time limit of one hour. However, the model terminates with a relative MIP (relMIP) gap around 0.1 (relMIP=$\frac{UB-LB}{UB}$, where UB is the best upper bound and LB is the LP relaxation-based lower bound obtained at termination).
Table 3: Solution time (in seconds) of model (S) for fully used providers

| Model (S) | $|I| = 6$ | $|P|$ | Min | Avg | Max | $|I| = 8$ | $|P|$ | Min | Avg | Max |
|-----------|------------|------|-----|-----|-----|------------|------|-----|-----|-----|
|           |            | 24   | 3.2 | 3.3 | 3.5 |            | 24   | 3.9 | 9.1 | 19.4|
|           |            | 30   | 7.6 | 9.1 | 13.1|            | 32   | 12.3| 13.3| 15.0|
|           |            | 36   | 18.6| 20.6| 22.3|            | 40   | 31.6| 46.4| 60.3|
|           |            | 42   | 99.5| 140.7|172.5|            | 48   | 462.2| 751.0|1288.8|
|           |            | 48   | 169.5|397.1|854.5|            | 56   | 1484.9|1848.6|3165.7|
|           |            | 54   | 68.1 |850.9|1821.7|            | 64   | -    |    |    |
|           |            | 60   | 532.3|1458.2|2435.1|            | 72   | -    |    |    |

When $|I| = 8, |P| \geq 32$, we set the relative MIP gap to 0.04.

In contrast, using model (Z), we were only able to solve small instances, specifically, all the SAAs that correspond to instances with $|I| = 6, |P| \leq 18$ and $|I| = 8, |P| \leq 16$. We present a comparison of solution times of these instances by model (S) and (Z) in Table 4. Clearly, model (Z) takes from 0.3 to 528 times longer than model (S) to solve these instances. Moreover, for those instances that model (Z) failed to solve within the time limit, it terminated with either a relMIP gap around 50% (when $|I| = 6$) or without any feasible MIP solutions (and thus no upper bound).

We attribute the difference in solution times to the following. First, as discussed in Section 5, model (Z) has significantly more binary first-stage variables than model (S). Moreover, model (Z) has a significantly larger number of first-stage constraints and a larger number of scenario-based constraints and variables. As argued in Klotz and Newman (2013), this increase in model size often suggests an increase in solution time for the LP relaxations. Second, as shown in Table 5, the LP relaxations obtained using model (S) were strictly tighter than using model (Z) by a factor of 1503 to 1789, which is consistent with the theoretical results in Theorem 2. Finally, model (S) for the fully used provider does not have constraints involving big-M coefficients, but model (Z) has such constraints. It is well-known that the big-M coefficients and constraints could undermine computational efficiency, enlarge the feasible region of the LP relaxation of the model, and cause numerical errors (Klotz and Newman, 2013; Camm et al., 1990). In Appendix G, we provide additional solution time results under a larger sample size and a different cost structure in the objective function. These results emphasize that model (S) is more computationally efficient than model (Z) under these settings.

It is worth noting that using off-the-shelf optimization software such as CPLEX to solve model (S) directly is more computationally efficient than using Benders decomposition (BD). Specifically, using BD, we were only able to solve small instances of HFASP with a maximum size of $(|I|, |P|)=(6, <24)$ within the given time limit. In contrast, model (S) can solve larger instances much faster, as shown in Table 3. Furthermore, when BD fails to solve instances within the given time limit, it terminates either with a large relMIP or without feasible MIP solutions. In such cases, the
Table 4: Ratios of solution times of models (Z) and (S) on the SAAs solved by both (fully used models).

| (Z) sol.time | (S) sol.time | \( |I| = 6 \) | \( |I| = 8 \) |
|--------------|--------------|-------------|-------------|
| | Min | Avg | Max | | Min | Avg | Max |
| 6 | 1.7 | 3.2 |
| 12 | 2.3 | 5.6 |
| 18 | 235.8 | 528.6 |

Table 5: Ratios of optimal objective values of LP relaxations of model (S) and (Z) (fully used models).

| (S) relaxation.obj | (Z) relaxation.obj | \( |I| = 6 \) | \( |I| = 8 \) |
|-------------------|-------------------|-------------|-------------|
| | Min | Avg | Max | | Min | Avg | Max |
| 6 | 1744.5 | 1750.8 | 1758.4 |
| 12 | 1761.4 | 1774.9 | 1789.6 |
| 18 | 1761.3 | 1563.1 | 1542.1 |

Table 6: Comparison of Solutions using Benders Decomposition (BD) and CPLEX (fully used models).

| \( (|I|, |P|) \) | Method | Sol.Time/Gap | Obj | Travel Time | Waiting Time | Overtime |
|----------------|--------|--------------|-----|-------------|--------------|---------|
| (6,30) | BD | 7.4% | 5454 | 821 | 2171 | 60 |
| Model (S) | 10.3 | 5083 | 819 | 4.6 | 0.1 |

solution obtained by BD at the termination has poor quality. For instance, consider the instance with \( (|I|, |P|) = (6,30) \). As shown in Table 6, BD fails to solve this instance within the given time limit and terminates with a 7% gap. Moreover, the solution obtained from BD at termination has longer travel time, waiting time, and overtime.

7.2.2. Partially used provider case

Let us now analyze solution times of the proposed models for the partially used provider case. We present results for problem instances with \( |I| = 6 \) (\( |I| = 8 \)) and \( |P| = \{6,8,\ldots,62\} \) (\( |P| = \{6,8,\ldots,42\} \)). We set the number of service providers \( |K| \) to \( \lceil |P|/6 \rceil + 1 \). We generated and solved five instances for each combination of \( |P| \) and \( |K| \). Table 7 presents the Min, Avg, and Max solution times of the instances solved by model (S). We observe that using model (S), we were able to solve all the instances with \( |I| = 6 \), \( |P| \leq 42 \) and \( |I| = 8 \), \( |P| \leq 30 \) within one minute. The average solution times using model (S) with \( |I| = 6 \) ranges from 6.8 seconds (\( |P| = 24 \)) to 6 minutes (\( |P| = 62 \)), and with \( |I| = 8 \) ranges from 11.8 seconds (\( |P| = 24 \)) to 18 minutes (\( |P| = 40 \)). Additionally, we could not solve problem instances with \( |I| = 8 \), \( |P| = 48 \) within one hour. In contrast, using model (Z), we were only able to solve small instances, namely all the SAAs that correspond to instances with \( |I| = 6 \) and \( |P| \leq 10 \) and with \( |I| = 8 \) and \( |P| \leq 8 \). We present a comparison of solution times of these instances by models (S) and (Z) in Table 8. Observe that the solution times of model (Z) is 2.4 to 6235.2 times longer than the solution times of model (S). Moreover, for those instances that
Table 7: Solution time (in seconds) of model (S) for partially used providers

| model (S) | $|I| = 6$ | $|I| = 8$ |
|-----------|-----------|-----------|
| $|P|$     | Min Avg Max | Min Avg Max |
| 24       | 6.1 6.8 7.4 | 9.2 11.8 15.5 |
| 30       | 15.1 16.1 18.6 | 43.6 50.5 55.1 |
| 36       | 17.0 18.8 20 | 168.6 191.5 216.3 |
| 40       | 27.4 29.6 32.2 | 644.5 1084.4 1819.4 |
| 42       | 28.8 31.6 34.0 | 484.7 738.6 967.6 |
| 48       | 51.8 61.8 72.6 | - - - |
| 50       | 67.5 109.8 148.2 | - - - |
| 54       | 93.0 102.0 120.2 | - - - |
| 58       | 122.4 199.7 306.7 | - - - |
| 60       | 148.8 175.3 253.4 | - - - |
| 62       | 154.2 361.5 1067.9 | - - - |

Table 8: Ratios of solution times of models (Z) and (S) on the SAAs solved by both (partially used provider models).

<table>
<thead>
<tr>
<th>(Z) sol.time</th>
<th>(S) sol.time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>Min Avg Max</td>
</tr>
<tr>
<td>6</td>
<td>4.2 4.5 4.9</td>
</tr>
<tr>
<td>8</td>
<td>81.5 88.2 102.8</td>
</tr>
<tr>
<td>10</td>
<td>3405.1 4286.9 6235.2</td>
</tr>
</tbody>
</table>

were not solved by model (Z) within the imposed time limit, the solver terminated with a relMIP of around 100% ($|I| = 6$) and 70% ($|I| = 8$) or without any feasible MIP solutions.

We attribute the differences in solution time to the following. First, model (Z) has more binary variables and constraints than model (S) (see Section 5), indicating that model (Z) is potentially more challenging to solve than model (S). Second, as shown in Table 9, the LP relaxations of model (S) are strictly tighter than that of model (Z), by a factor of 1500.6 to 1771 (proved theoretically in Theorem 2). We also compare the computational performance of models (S) and (Z) using another cost structures and a larger number of scenarios (see Appendix H for details). We observe that model (S) is always better than model (Z) in the sense that model (S) can solve larger instances and is more computationally efficient.

Finally, it is worth noting that using BD, we cannot solve even small instances with 12 customers and 3 providers within the imposed time limit. For example, after two hours, the average BD gap for instances with $(|I|, |P|, |K|) = (6, 12, 3)$ is approximately 6%. In contrast, model (S) can solve this instance within a few seconds and quickly solve other larger instances (see Table 7). This again emphasizes that solving model (S) directly is more computationally efficient.
Table 9: Ratios of optimal objective values of LP relaxations of model (S) and (Z) (partially used provider models).

<table>
<thead>
<tr>
<th>(S) relaxation.obj</th>
<th>(Z) relaxation.obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>6</td>
<td>1744.5</td>
</tr>
<tr>
<td>8</td>
<td>1745.0</td>
</tr>
<tr>
<td>10</td>
<td>1760.1</td>
</tr>
</tbody>
</table>

Table 10: The relative gap between $\nu^*$ and $\nu$

| $|I|=6$           | $|I|=8$           |
|------------------|------------------|
| $|P|$ | FAS-RS | FAS-R-S | $|P|$ | FAS-RS | FAS-R-S |
| 24  | 0.89% | 1.21% | 24  | 1.62% | 3.43% |
| 30  | 0.96% | 1.36% | 32  | 2.03% | 3.58% |
| 36  | 0.99% | 1.31% | 40  | 1.74% | 5.10% |
| 42  | 1.01% | 1.16% | 48  | 1.90% | 4.34% |
| 48  | 1.01% | 1.14% | 56  | 2.11% | 3.50% |
| 54  | 1.01% | 1.12% | 64  | 2.00% | 2.85% |

| $|I|=6$           | $|I|=8$           |
|------------------|------------------|
| $|P|$ | FAS-R-S | FAS-R-S | $|P|$ | FAS-R-S | FAS-R-S |
| 24  | 1.05% | 1.16% | 24  | 2.08% | 3.52% |
| 36  | 1.20% | 1.29% | 36  | 2.02% | 5.07% |
| 42  | 1.15% | 1.07% | 42  | 1.93% | 4.39% |
| 48  | 1.15% | 1.21% | 48  | 1.15% | 3.92% |
| 54  | 1.15% | 1.12% | 54  | 1.15% | 3.09% |
| 60  | 1.15% | 1.07% | 60  | 1.15% | 4.23% |

7.3. Analysis of the FAS-RS and FAS-R-S Heuristics

In this section, we investigate solutions quality and computational performance of the FAS-RS and FAS-R-S heuristics. First, in Table 10, we present the relative gap $\frac{\nu^* - \nu}{\nu^*} \times 100\%$ between the optimal objective value $\nu^*$ of model (S) and the objective value $\nu$ computed using solutions obtained via the proposed heuristics for all instances that model (S) can solve to optimality. The small gap values in Table 10 indicate that FAS-RS and FAS-R-S could produce near-optimal solutions to the HFASP with gap values ranging from 0.86% to 5.10%. However, FAS-RS solutions generally yield lower gap values than FAS-R-S, especially when $|I|=8$. This makes sense because FAS-RS employs model (S) in phase 2 to optimize routing and scheduling decisions for each provider. In contrast, FAS-R-S uses the MIH heuristic to obtain routing decisions in phase 2 and then an LP to find the appointment times in phase 3.

Next, we investigate the computational performance of FAS-RS and FAS-R-S on solving large instances that might be observed in other application domains. Table 11 presents the total time required by each heuristic to solve large instances with customers ranging from 102 to 504 (fully used) and 100 to 500 (partially used). We observe the following from this table. First, both heuristics can efficiently solve large instances of the HFASP within a reasonable time. Specifically,
Table 11: Solution times (in seconds) using FAS-RS and FAS-R-S

<table>
<thead>
<tr>
<th>Fully used Provider Case</th>
<th></th>
<th>Partly used Provider Case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>$</td>
<td>P</td>
</tr>
<tr>
<td>102</td>
<td>2.8</td>
<td>2.8</td>
</tr>
<tr>
<td>204</td>
<td>22.8</td>
<td>22.8</td>
</tr>
<tr>
<td>300</td>
<td>31.8</td>
<td>31.8</td>
</tr>
<tr>
<td>402</td>
<td>46.4</td>
<td>46.4</td>
</tr>
<tr>
<td>504</td>
<td>64.4</td>
<td>64.4</td>
</tr>
</tbody>
</table>

solution times range from 2.8 seconds (using FAS-RS for $|P| = 104$ and $|I| = 8$ fully used provider case) to 16 minutes (using FAS-RS for $|P| = 504$ and $|I| = 8$ fully used provider case). Second, the solution times of FAS-R-S are shorter than FAS-RS, and the difference is significant for larger instances with $I = 8$. For example, consider the fully used provider case with $|I| = 8$ and $|P| = 504$. FAS-R-S takes about 34 seconds to solve this instance, while FAS-RS takes 16 minutes. Recall that FAS-RS involves solving a SMIP in phase 2 to obtain routing and scheduling decisions. In contrast, FAS-R-S obtains routes using the MIH heuristic and then solves an LP to obtain appointment times. Thus, it makes sense that FAS-RS takes a longer time to solve each instance.

Note that using model (S) in phase 2 of FAS-RS is more computationally efficient than using model (Z). For example, consider the partially used provider case. For example, solution times of instances with $|P| = (100, 200, 300, 400)$ and $I = 8$ using FAS-RS with model (Z) implemented in phase 2 are (470, 734, 1480, 1683) seconds and with model (S) implemented in phase 2 are (116, 197, 453, 596). These results demonstrate that model (S) also enables computationally efficient heuristics for the HFASP.

7.4. Case Study

In this section, we consider a service region based on twenty-five cities in Lehigh County of Pennsylvania (see Figure 2). Then, we construct two HFASP instances based on this region as follows. First, we used the population estimate for each city based on the most updated information posted in 2013-2017 US Census Bureau to construct two instances denoted as L-50 and L-100, where 50 and 100 are the total number of customers. In L-50 and L-100, we use the population percentage (weight) in each city to calculate the number of customers as population% × 50 and
population\% \times 100$, respectively (see Table J1 in Appendix J). To a certain extent, these instances reflect what may be observed in real life, i.e., locations with larger populations may potentially create greater demand. We chose Lehigh Valley Hospital Home Care (in Allentown), which primarily serves Lehigh County, as the depot (provider office). For each instance, we first randomly located customers (nodes) in each city at some residential area within the city, such as apartments or gated communities (see Figure J1 in Appendix J). Second, we obtained the travel time $\bar{t}_{p,q}$ between each pair of customers $(p, q)$ and used it as the average travel time. Third, we generated the non-negative travel time between each pair of customers $(p, q)$ from a normal distribution $N(\bar{t}_{p,q}, \bar{t}_{p,q}/6)$. Finally, we follow the same procedure described in Section 7.1 to generate the service time and other parameters. Next, we analyze the impact of key input parameters on the optimal solutions and the associated operational performance.

7.4.1. Impact of variability in service time

In this section, we analyze the impact of service time variability on the number of hired providers and the associated operational (second-stage) cost. In addition to the base range of average service time ($\mu \sim U[25,35]$), we consider the following three ranges: (a) $\mu \sim U[25,50]$; (b) $\mu \sim U[50,60]$; and (c) $\mu \sim U[50,90]$. In ranges (a) and (c), we increase the variability of service time by extending the range (difference between the lower and upper bounds) of $\mu$ from 10 to 25 and 40, respectively. In range (b), we keep the difference between the upper and lower bounds of $\mu$ to 10 and increase the average length of service time. Note that ranges (b) and (c) correspond to applications with longer service time than the base case and range (a). In addition, we consider fixed cost $\lambda^f \in \{50,100,1000\}$, $|I| = 6$, and $|K| = 24$ and 45 for L-50 and L-100 instances, respectively. All other
The number of hired providers and associated second stage cost under different $\lambda^f$ and range of $\mu$.

Figure 3 shows the number of hired providers and the second-stage operational cost under each combination of $\mu$ and $\lambda^f$. We observe the following from this figure. First, we need more providers to serve all customers in the L-100 instance under most ranges of $\mu$ and values of $\lambda^f$, which makes sense as we have a larger number of customers in this instance. Second, the optimal number of hired providers under the base range and range (a) equals the minimum number of providers required to serve the customers (9 and 18 providers for L-50 and L-100, respectively). This makes sense because under these ranges, the length and variability of service time are lower than the remaining ranges, and both the waiting time and overtime values are low (see Figures 3b and 3d). Thus, hiring additional providers will not improve the second-stage objective but will increase the fixed cost. Third, the optimal number of hired providers under ranges (b) and (c) (i.e., longer service time) with $\lambda^f \in \{50, 100\}$ (i.e., lower fixed cost) is larger than the base range and range (a). Moreover, the optimal number of hired providers under range (c) with $\lambda^f \in \{50, 100\}$ is larger than under range (b). These results make sense because the service time is longer under (c) and (b), and thus, we need more providers to serve customers and mitigate overtime and waiting time. Finally, the
optimal number of hired providers under a large value of $\lambda^f = 1000$ equals the minimum number of the required providers to serve the 50 customers under all service time ranges. This is because when $\lambda^f = 1000$, the fixed cost is much higher than the operational cost. However, by hiring fewer providers under $\lambda^f = 1000$, the overtime and waiting time worsen, especially under ranges (b) and (c) (see Figures 3b and 3d).

7.4.2. Impact of variability in the unit waiting and overtime costs

In this section, we analyze the optimal number of hired providers as a function of unit waiting time and overtime costs. First, we fix $(\lambda^f, \lambda^t, \lambda^o) = (100, 0.1, 1)$, and vary $\lambda^w \in \{1, 5, 10, 15, 20, 25\}$. For brevity, we discuss results for L-100, and the observations for L-50 are similar. Figure 4 shows the optimal number of hired providers and the associated total waiting time for each combination of $\lambda^w$ and $\mu$. We observe the following from this figure. First, the optimal number of hired providers under the base range is 18 (i.e., the minimum number of providers required to serve all the 100 customers) irrespective $\lambda^w$. This is reasonable because the service time under the base range is lower than the other ranges. Moreover, as shown in Figure 4b, the associated total waiting time under this range is very low. Second, the optimal number of hired providers increases with $\lambda^w$ under ranges (a)–(c), consequently leading to lower waiting times. These results make sense because the length and variability of service time under ranges (a)–(c) is greater than the base range. Thus, by hiring more providers, we could mitigate excessive waiting time.

Next, we analyze the impact of the unit overtime cost. Figure 5 shows the optimal number of hired providers and total overtime for each combination of $\lambda^o$ and $\mu$. We observe that the optimal number of hired providers under the base range and range (a) is 18 (i.e., the minimum number of providers required to serve all the 100 customers) irrespective of $\lambda^o$. This is because the total overtime under the base range and range (a) is small (see Figure 5b). On the other hand, the optimal number of hired providers under range (c) is greater than the remaining ranges. This is reasonable because the length and variability of service time under range (c) are larger than the remaining ranges, and additional providers are needed to mitigate providers’ overtime.
8. Conclusion

In this paper, we propose and analyze two new SMIPs, denoted as model (S) and model (Z), for the HFASP with random service and travel times. Specifically, given sets of providers and customers, these models aim to find the number of providers to hire, the order of customers assigned to each provider, and an appointment time for each customer. Given the uncertainty in service and travel times, the goal is to minimize the sum of the fixed cost of hiring providers plus the expected cost associated with customers’ waiting time and providers’ travel time, overtime, and idle time.

The HFASP is an important multiple-vehicle fleet sizing, routing, and scheduling problem that has been studied in closely related contexts with random service time and deterministic travel times. Therefore, in model (Z), we extend an existing model that employs traditional routing variables and constraints for a closely related problem by incorporating the co-existing uncertainty of random travel and service times and providers’ idle time. In model (S), we propose a new sequencing-based formulation of the problem. Our theoretical analyses show that model (S) is smaller and provides a tighter LP relaxation, suggesting a better computational performance. Indeed, the computational results demonstrate that significant improvements in computational performance could be gained with model (S) over model (Z). We also propose two computationally efficient heuristics and show that they could quickly obtain near-optimal solutions to large instances of the problem. Finally, we use instances based on Lehigh County of PA to derive insights into the HFASP.

We suggest the following areas for future research. First, our proposed models can be considered the first step towards building comprehensive stochastic optimization approaches for home service staffing, capacity planning, and routing and scheduling, considering all relevant organizational and technical constraints. Second, in these extensions, one could consider various sources of uncertainty, such as random demand, capacities, and cancellations. Finally, designing user-friendly decision support tools that implement the models will enable practitioners to adopt them in practice.


Appendix A. Model (S) Derivation

We show the detailed derivation of model (S). We start with the second-stage formulation without the travel cost. For notational convenience, we suppress the scenario index n ∈ [N] from the scenario-dependent variables and parameters. For all i ∈ I, we define the actual arrival time of the ith appointment by $R_i$. It is clear that $R_i$ should satisfy

$$R_1 = \sum_{p \in P} t_{0,p} x_{1,p},$$

$$R_i = \begin{cases} S_{i-1} + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1,p_2) \in P \times P} t_{p_1,p_2} z_{i,p_1,p_2}, & \text{if } \sum_{p \in P} x_{i,p} = 1, \\ 0, & \text{if } \sum_{p \in P} x_{i,p} = 0, \end{cases} \quad \forall i \in [2, |I|].$$

Next, we define the actual start time of the ith appointment by $S_i$. Note the ith appointment cannot start before the scheduled appointment time, $a_i$, nor the provider’s actual arrival time $R_i$. Mathematically, the actual start time of the ith appointment $S_i$ should satisfy

$$S_i = \max\{R_i, a_i\}, \quad \forall i \in I.$$

If the completion time of all appointments assigned to a provider exceeds the working hours, s/he experiences overtime. We let $O$ represent the provider’s overtime, and compute it as follows.

$$O = \left( \max_{i \in I} \left\{ S_i + \sum_{p \in P} (d_p + t_{p,0}) x_{0,p} - L \right\} \right)^+, \tag{A.1a}$$

where, $(b)^+ = \max\{b, 0\}$. If the provider arrives before the appointment time, he/she must wait (i.e., remain idle) until the scheduled appointment time to start the service. Mathematically, we can compute the provider idle time cost before the ith appointment as $\lambda^g(a_i - R_i)^+$. If the provider arrives after the scheduled appointment time, the customer experience waiting. Mathematically, we can compute the waiting cost as $\lambda^w(R_i - a_i)^+$, for all $i \in I$. Accordingly, the second-stage formulation of the HFASP without the travel cost is as follows.

$$(P0) \min_{S,R,O} \lambda^w \sum_i (R_i - a_i)^+ + \lambda^g \sum_i (a_i - R_i)^+ + \lambda^o O \tag{A.1a}$$

s.t.  \quad S_i = \max\{R_i, a_i\}, \quad \forall i \in I, \tag{A.1b}

$$R_1 = \sum_{p \in P} t_{0,p} x_{1,p}, \tag{A.1c}$$
\[
R_i = \left[ S_{i-1} + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1, p_2) \in P \times P} t_{p_1, p_2} z_{i, p_1, p_2} - M \left( 1 - \sum_{p \in P} x_{i,p} \right) \right]^+, \quad \forall i \in [2, |I|],
\]
\[\text{(A.1d)}\]

\[
O = \left( \max_{i \in I} \left\{ S_i + \sum_{p \in P} (d_p + t_{p,0}) x_{0,p} - L \right\} \right)^+.
\]
\[\text{(A.1e)}\]

Note that formulation (A.1) is not straightforward to solve in its presented form. Next, in Theorem 3, we derive an equivalent mixed integer program reformulation that is solvable.

**Theorem 3.** Problem \((P0)\) is equivalent to the following mixed integer programming model \((P2)\).

\[
(P2) \quad \min_{s, g, o} \lambda^w \sum_{i \in I} (s_i - a_i) + \lambda^g \sum_{i \in I} g_i + \lambda^o o
\]
\[\text{(A.2a)}\]

\[\text{s.t.}\]
\[
s_i \geq a_i, \quad \forall i \in I,
\]
\[\text{(A.2b)}\]

\[
s_1 \geq \sum_{p \in P} t_{0,p} x_{1,p},
\]
\[\text{(A.2c)}\]

\[
s_i \geq s_{i-1} + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1, p_2) \in P \times P} t_{p_1, p_2} z_{i, p_1, p_2} - M \left( 1 - \sum_{p \in P} x_{i,p} \right), \forall i \in [2, |I|],
\]
\[\text{(A.2d)}\]

\[
g_1 \geq s_1 - \sum_{p \in P} t_{0,p} x_{1,p},
\]
\[\text{(A.2e)}\]

\[
g_i \geq s_i - \left( s_{i-1} + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1, p_2) \in P \times P} t_{p_1, p_2} z_{i, p_1, p_2} \right), \quad \forall i \in [2, |I|],
\]
\[\text{(A.2f)}\]

\[
o \geq s_i + \sum_{p \in P} (d_p + t_{p,0}) x_{i,p} - L, \quad \forall i \in I,
\]
\[\text{(A.2g)}\]

\[
(s, g, o) \geq 0.
\]
\[\text{(A.2h)}\]

**Proof of Theorem 3.** Observe that, for all \(i \in I\), we have \(S_i = \max\{R_i, a_i\} = (R_i - a_i)^+ + a_i = (a_i - R_i)^+ + R_i\). Thus, for a feasible first-stage solution \((x, a)\), the objective function of \((P0)\) equals

\[
\lambda^w \sum_{i \in I} (S_i - a_i) + \lambda^g \sum_{i \in I} (S_i - R_i) + \lambda^o \left( \max_{i \in I} \left\{ S_i + \sum_{p \in P} (d_p + t_{p,0}) x_{0,p} - L \right\} \right)^+
\]

\[
= \lambda^w \sum_{i \in I} (S_i - a_i) + \lambda^g \left( S_1 - \left[ \sum_{p \in P} t_{0,p} x_{1,p} \right] \right)
\]

\[
+ \lambda^g \sum_{i=2}^{|I|} \left( S_i - \left[ S_{i-1} + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1, p_2) \in P \times P} t_{p_1, p_2} z_{i, p_1, p_2} - M \left( 1 - \sum_{p \in P} x_{i,p} \right) \right] \right)^+
\]

\[
+ \lambda^o \left( \max_{i \in I} \left\{ S_i + \sum_{p \in P} (d_p + t_{p,0}) x_{0,p} - L \right\} \right)^+.
\]

Hence, \((P0)\) is equivalent to

\[
(P0') \quad \min_{s} \lambda^w \sum_{i \in I} (s_i - a_i) + \lambda^g \left( s_1 - \left[ \sum_{p \in P} t_{0,p} x_{1,p} \right] \right)
\]

32
Since the objective function (A.3a) is increasing in \( v \) is the big-M constant in constraints (A.5c). To show the equivalence, we let \( v_1 \) and \( v_2 \) represent the optimal values of (P1) and (P2), respectively. We claim that \( v_1 = v_2 \). From our derivation of (P1), we know that an optimal solution of (P1) is given by \( g^* \).
\[ g_i^* = s_i^* - \left[ s_{i-1}^* + \sum_{p \in P} d_p x_{i-1,p} + \sum_{(p_1,p_2) \in P \times P} t_{p_1,p_2} z_{i,p_1,p_2} - M \left( 1 - \sum_{p \in P} x_{i,p} \right) \right]^+, \quad \forall i \in [2, |I|], \]
\[ o^* = \left( \max_{i \in I} \left\{ s_i^* + \sum_{p \in P} (d_p + t_{p,0}) x_{i,p} - L \right\} \right)^+, \]
where \( s_i^* \) is the actual appointment start time of \( i \)th appointment if \( \sum_{p \in P} x_{i,p} = 1 \), and is zero otherwise, for all \( i \in I \). It is easy to verify that \( g^* \) satisfies (A.2e)–(A.2f) and thus, this optimal solution to (P1) is feasible to (P2). For a minimization problem, a feasible solution is always an upper bound for the optimal solution. Since the objective functions of (P1) and (P2) are the same, we must have \( v_2 \leq v_1 \). On the other hand, constraints (A.5c) is a relaxation of (A.2f) because of the existence of the big-M constant, i.e., the feasible region of (P1) contains that of (P2). Hence, we also have \( v_1 \leq v_2 \). Therefore, we conclude that \( v_1 = v_2 \), implying that (P1) is equivalent to (P2). This completes the proof. \( \square \)

Appendix B. Proof of Proposition 1

Proof. Consider any provider \( k \in K \) and position \( i \in [2, |I|] \) in his/her sequence of customers. From constraints (1l), we have
\[ s_{n,i,k} \geq s_{i-1,k} + \sum_{p \in P} d_p x_{i-1,p,k} + \sum_{(p,q) \in P \times P, p \neq q} t_{p,q} z_{i,p,q,k} - M_i \left( 1 - \sum_{p \in P} x_{i,p,k} \right). \]
It follows that to preserve optimality when \( \sum_{p \in P} x_{i,p,k} = 0 \), \( M_i \) should be
\[ M_i \geq s_{i-1,k} + \sum_{p \in P} d_p x_{i-1,p,k} + \sum_{(p,q) \in P \times P, p \neq q} t_{p,q} z_{i,p,q,k} - s_{i,k}, \]
\[ \geq s_{i-1,k} + \sum_{p \in P} d_p x_{i-1,p,k} + \sum_{(p,q) \in P \times P, p \neq q} t_{p,q} z_{i,p,q,k} \quad \text{(since } s_{i,k}^n \geq 0). \]
By constraints (1k) and (1l), we can recursively find that
\[ M_i \geq \max_{k \in K, n \in [N], i \in [2, |I|]} \left\{ s_{n,1,k} + \sum_{j=1}^{i-1} \sum_{p \in P} d_p x_{j,p,k} + \sum_{j=2}^{i} \sum_{(p,q) \in P \times P, p \neq q} t_{p,q} z_{j,p,q,k} \right\}. \quad (B.1) \]
Since \((\lambda^w, \lambda^g, \lambda^o) > 0\), the actual start time of the 1st customer \( s_{1,k}^n \) is at most \( L + t_1^{\text{max}} \) by constraints (1i) and (1k), for any scenario \( n \in [N] \) and provider \( k \in K \). It follows from (B.1) that
\[ M_i \geq L + t_1^{\text{max}} + (i - 1)d + (i - 1)\bar{t}, \quad \forall i \in [2, |I|]. \quad (B.2) \]
\( \square \)

Appendix C. Proof of Proposition 2

Proof. Let us first prove the validity of \( M \) in constraints (3k), (3m), and (3n) of model (Z). For constraints (3k) and (3n), we consider the following two cases:
\[ S^n_q \geq S^n_0 + d^n_0 + t^n_{0,q} - M(1 - \sum_{k \in K} z_{0,q,k}), \quad \forall n \in [N] \tag{C.1a} \]

\[ S^n_q \geq t^n_{0,q} - M\left(1 - \sum_{k \in K} z_{0,q,k}\right), \quad \forall n \in [N]. \tag{C.1b} \]

Inequalities (C.1b) hold because (1) \( S^n_0 = 0 \) by constraints (3f), and (2) \( d^n_0 = 0 \). When \( \sum_{k \in K} z_{0,q,k} = 0 \), to ensure feasibility, we should satisfy
\[ M \geq t^n_{0,q} \quad \forall n \in [N]. \tag{C.2} \]

Similarly, by constraints (3n), we have
\[ M \geq S^n_q - S^n_0 - d^n_0 - t^n_{0,q} - G^n_q \geq S^n_q - t^n_{0,q}, \quad \forall n \in [N]. \tag{C.3} \]

• Case 2: \( p \in P, q \in P \). If \( \sum_{k \in K} z_{p,q,k} = 0 \). By constraints (3k), we have
\[ M \geq S^n_p + d^n_p + t^n_{p,q} - S^n_q \geq S^n_p + d^n_p + t^n_{p,q}, \quad \forall n \in [N]. \tag{C.4} \]

By constraints (3n), we have
\[ M \geq S^n_q - S^n_p - d^n_p - t^n_{p,q} - G^n_q \geq S^n_q - (S^n_p + d^n_p + t^n_{p,q}), \quad \forall n \in [N]. \tag{C.5} \]

It is easy to verify from constraints (3m) that \( M \) should satisfies:
\[ M \geq S^n_p + d^n_p + t^n_{p,0} - L - O^n_k \geq S^n_p + d^n_p + t^n_{p,0}, \quad \forall n \in [N]. \tag{C.6} \]

Combining inequalities (C.2)–(C.6), we conclude that \( M \) should satisfies:
\[ M \geq \max_{p \in P, n \in [N]} \{S^n_p\} + \bar{d} + t^{\text{max}}_2. \tag{C.7} \]

Next, we derive an upper bound on the actual start time. From constraints (3f), we know that each provider will serve at most \(|I|\) customers. Moreover, the scheduled appointment time in any feasible solution should be less than or equal to \( L \) by constraints (3g). Thus, when \( \lambda^o > 0 \) and \( \lambda^w > 0 \), the actual start time of the first customer \( p' \in P \) in the sequence of customers assigned to any provider \( k \in K \) (i.e., \( z_{o,p',k} = 1 \)) should satisfies: \( S^n_{p'} \leq L + t^{\text{max}}_1. \) Moreover, it is easy to verify that the actual start time of any customer \( p \in P \) satisfies:
\[ S^n_p \leq L + t^{\text{max}}_1 + (|I| - 1)(\bar{d} + \bar{l}), \quad \forall n \in [N]. \tag{C.8} \]

It follows from inequalities (C.7) and (C.8) that to preserve optimality, it is sufficient to choose the Big-M constant as \( M \geq L + t^{\text{max}}_1 + |I|\bar{d} + (|I| - 1)\bar{l} + t^{\text{max}}_2 \). This completes the proof. \( \square \)

Appendix D. Proof of Theorem 1

Proof. Suppose \((\bar{x}, \bar{z}, \bar{a}, \bar{s}, \bar{g}, \bar{o})\) is an optimal solution to model (S). Below, we construct a feasible solution to model (Z) with the same objective function value.

A) For all \( p \in P, k \in K \), let \( z_{0,p,k} = \bar{x}_{1,p,k} \) and \( z_{p,0,k} = \bar{x}_{|I|,p,k} \). In addition, let \( z_{0,0,k} = 1 - \sum_{p \in P} \bar{x}_{1,p,k} \). For a fixed \( k \in K \), note that if \( \sum_{p \in P} \bar{x}_{1,p,k} = 0 \) (i.e., provider \( k \) is not hired in the optimal solution to model (S)), we have \( \bar{x}_{i,p,k} = 0 \) for all \( i \in I \) and \( p \in P \), by constraints

35
D) Now it remains to show that \( z_{|I|,p',k} = 1 \), then by constraints (2c), there must exist \( p' \in P \) such that \( \bar{x}_{|I|,p',k} = 1 \). Therefore, we conclude that

\[
\sum_{p \in P \cup \{0\}} z_{0,p,k} = \sum_{p \in P} z_{0,p,k} + z_{0,0,k} = \sum_{p \in P} \bar{x}_{1,p,k} + \left(1 - \sum_{p \in P} \bar{x}_{1,p,k}\right) = 1,
\]

\[
\sum_{p \in P \cup \{0\}} z_{p,0,k} = \sum_{p \in P} z_{p,0,k} + z_{0,0,k} = \sum_{p \in P} \bar{x}_{|I|,p,k} + \left(1 - \sum_{p \in P} \bar{x}_{1,p,k}\right) = 1.
\]

Thus \( z_{0,p,k} \) and \( z_{0,0,k} \) satisfy constraints (3e) and (3d), respectively.

B) Let \( z_{p,q,k} = \sum_{i=2}^{|I|} (\bar{x}_{i-1,p,k} \cdot \bar{x}_{i,q,k}) \) for all \( p \in P, q \in P \) and \( k \in K \). We first show that \( z_{p,q,k} \leq 1 \) and hence, binary. Since \( \bar{x} \) is binary and satisfies constraints (1b) (i.e., for each \( p \in P \), there exists a unique pair \((i,k) \in I \times K \) such that \( \bar{x}_{i,p,k} = 1 \)), we have \( \bar{x}_{i_1,p,k} \in \{0,1\}, \bar{x}_{i_2,q,k} \in \{0,1\} \), where \( i_1, i_2 \subset I \) with \( i_1 \neq i_2 \). It follows that \( z_{p,q,k} = \sum_{i=2}^{|I|} (\bar{x}_{i-1,p,k} \cdot \bar{x}_{i,q,k}) \) is binary, for all \( p \in P, q \in P \). Next, we show that \( z_{p,q,k} \) satisfies constraints (3b). Since \( \bar{x} \) satisfies constraint (2c), we know that, for each \((i,k) \in I \times K \), there exists a \( q \in P \) such that \( \bar{x}_{i,q,k} = 1 \). Thus

\[
\sum_{k \in K} \sum_{q \in P \cup \{0\}} z_{p,q,k} = \sum_{k \in K} \left( \sum_{q \in P} z_{p,q,k} + z_{p,0,k} \right)
\]

\[
= \sum_{k \in K} \sum_{q \in P} \sum_{i=2}^{|I|} (\bar{x}_{i-1,p,k} \bar{x}_{i,q,k} + \bar{x}_{|I|,p,k})
\]

\[
= \sum_{k \in K} \left( \sum_{i=2}^{|I|} \bar{x}_{i-1,p,k} + \bar{x}_{|I|,p,k} \right) \quad \text{(since } \bar{x} \text{ satisfies constraints (3b))}
\]

\[
= \sum_{k \in K} \sum_{i \in I} \bar{x}_{i,p,k} = 1, \quad \text{(since } \bar{x} \text{ satisfies constraints (1b))}
\]

which implies that \( z_{p,q,k} \) satisfies constraints (3b).

C) Next, we show that \( z \) satisfies constraints (3c). Since \( \bar{x} \) satisfies constraints (2c), for each \((i,k) \in I \times K \), there must exists \( p \) such that \( \bar{x}_{i,p,k} = 1 \). Thus, for a fixed \((q,k) \), we have

\[
\sum_{p \in P \cup \{0\}} z_{p,q,k} - \sum_{p \in P \cup \{0\}} z_{q,p,k} = \left[ \sum_{p \in P} z_{p,q,k} + z_{0,q,k} \right] - \left[ \sum_{p \in P} z_{q,p,k} + z_{0,q,k} \right]
\]

\[
= \left[ \sum_{p \in P} \sum_{i=2}^{|I|} (\bar{x}_{i-1,p,k} \bar{x}_{i,q,k}) + \bar{x}_{1,q,k} \right] - \left[ \sum_{p \in P} \sum_{i=2}^{|I|} (\bar{x}_{i-1,q,k} \bar{x}_{i,p,k}) + \bar{x}_{|I|,q,k} \right]
\]

\[
= \left[ \sum_{i=2}^{|I|} \bar{x}_{i,q,k} + \bar{x}_{1,q,k} \right] - \left[ \sum_{i=2}^{|I|} \bar{x}_{i-1,q,k} + \bar{x}_{|I|,q,k} \right] = 1 - 1 = 0.
\]

Thus \( z_{p,q,k} \) satisfies constraints (3c).

D) Now it remains to show that \( z_{p,q,k} \) satisfies constraints (4a) for all \( p \in P \cup \{0\}, q \in P \cup \{0\} \) and \( k \in K \). For a fixed \( k \in K \), by definition of \( z_{p,q,k} \) in points (A) and (B), we have

\[
\sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} z_{p,q,k} = \sum_{p \in P} \sum_{q \in P} z_{p,q,k} + \sum_{p \in P} z_{p,0,k} + \sum_{q \in P} z_{0,q,k}
\]
Let \( p \in P \) and \( q \in P \). Since \( \bar{a}_{i,k} \) satisfies constraints (1b), by constraints (2d), we have
\[
A_p = \sum_{i \in I} \sum_{k \in K} \bar{a}_{i,k} \bar{x}_{i,p,k} \leq L \sum_{i \in I} \sum_{k \in K} \bar{x}_{i,p,k} = L, \quad \forall p \in P.
\]
Thus, \( A_p \) satisfies constraints (3g).

F) Let \( S_p^n = \sum_{i \in I} \sum_{k \in K} \bar{s}_{i,k} \bar{x}_{i,p,k} \) for all \( p \in P \) and \( n \in [N] \). Since \( \bar{s}_{i,k} \) satisfies constraints (1j), by constraints (1b) and point (E), we have
\[
S_p^n = \sum_{i \in I} \sum_{k \in K} \bar{s}_{i,k} \bar{x}_{i,p,k} \geq \sum_{i \in I} \sum_{k \in K} \bar{a}_{i,k} \bar{x}_{i,p,k} = A_p.
\]
Thus, \( S_p^n \) satisfies constraints (3j).

G) To verify that \( S_p^n \) satisfies constraints (3k), for each \( n \in [N] \), we consider the following cases.

- Case 1: \( p = 0, q \in P \). In this case, let \( S_0^n = d_0^n = 0 \), for a fixed \( q \in P \), constraints (3k) are equivalent to
\[
S_q^n \geq S_0^n + d_0^n + t_{0,q}^n - M(1 - \sum_{k \in K} z_{0,q,k}) = t_{0,q}^n - M(1 - \sum_{k \in K} z_{0,q,k}).
\]
If \( \bar{x}_{1,q,k} = 0 \) for all \( k \in K \), then by point (A), we have \( z_{0,q,k} = 0 \) for all \( k \in K \). Thus, constraints (3k) are relaxed with a sufficiently large number \( M \) because \( \sum_{k \in K} z_{0,q,k} = 0 \). One the other hand, if \( \bar{x}_{1,q,k} = 1 \) for some \( \hat{k} \in K \) then \( z_{0,q,k} = \sum_{k \in K} z_{0,q,k} = 1 \) (by point (A)). From the definition of \( S_q^n \) in point (F), we have \( S_q^n = \bar{s}_{1,\hat{k}}^n \). Then we have
\[
S_q^n - S_p^n = S_q^n - S_0^n = \bar{s}_{1,\hat{k}}^n - 0
\]
\[
\geq \sum_{p' \in P} t_{0,q,p'}^n \bar{x}_{1,p',k} \quad \text{(since \( \bar{s}_{1,\hat{k}}^n \) satisfies constraints (1k))}
\]
\[
= t_{0,q}^n \quad \text{(since \( \sum_{p' \in P} \bar{x}_{1,p',k} = \bar{x}_{1,q,k} = 1 \))}
\]
- Case 2: Consider the case when \( (p, q) \in P \times P \). Without loss of generality, we assume that \( p \neq q \). By constraints (1b), there must exist \( \bar{x}_{i_1,p,k_1} = \bar{x}_{i_2,q,k_2} = 1 \), for some \( \{i_1, i_2\} \in I \) and \( \{k_1, k_2\} \in K \). By definition of \( z_{p,q,k} \) in point (B), it is trivial that if
H) Let \( \bar{z}_{i,p,q,k} = 0 \), and constraints (3k) are relaxed. Now, consider the first case when \( \bar{x}_{i-1,p,k} = \bar{x}_{i,q,k} = 1 \) for some \( i \in [2, |I|] \) and \( k \in K \). By constraints (1f)-(1h), we have \( \bar{z}_{i,p,q,k} = \bar{z}_{i-1,p,k} \bar{z}_{i,q,k} = 1 \). Accordingly, by definition of \( S \) in point (F), we have \( S_{p}^{n} = S_{p}^{n-1} \). Hence, \( S_{p}^{n} - S_{p}^{n-1} = d_{p}^{n} + t_{p,q}^{n} \).

I) Let \( \bar{O}_{k}^{n} = \bar{o}_{k}^{n} \) for all \( k \in K \) and \( n \in [N] \). For a fixed \( k \in K \), if \( \bar{x}_{|I|,p,k} = z_{p,0,k} = 1 \) for some \( p \in P \). Then, by constraints (1o), for each \( n \in [N] \) and for a fixed \( i \in I \), we have

\[
\bar{O}_{k}^{n} = \bar{o}_{k}^{n} \geq \bar{s}_{i,k}^{n} + \sum_{p \in P} (d_{p}^{n} + t_{p,0}^{n}) \bar{x}_{|I|,p,k} - L
\]

\[
\geq \bar{s}_{i,k}^{n} + d_{p}^{n} + t_{p,0}^{n} - L \quad (\text{since } \sum_{p \in P} \bar{x}_{|I|,p,k} = \bar{x}_{|I|,p,k} = 1) \quad (D.1)
\]

Consider the inequality (D.1) when \( i = |I| \). By definitions of \( S \) in point (F), we have \( \bar{s}_{n,k}^{n} = S_{p}^{n} \). Therefore, inequality (D.1) is equivalent to

\[
\bar{O}_{k}^{n} = \bar{o}_{k}^{n} \geq \bar{S}_{p}^{n} + d_{p}^{n} + t_{p,0}^{n} - L
\]

On the other hand, if \( \bar{x}_{|I|,p,k} = z_{p,0,k} = 0 \) for all \( p \in P \), constraints (1o) are relaxed to \( \bar{o}_{k}^{n} \geq 0 \) with the non-negativity of the decision variable \( \bar{O} \). Since \( z_{p,0,k} = 0 \) for all \( p \in P \), constraints (3m) are also relaxed to \( \bar{O}_{k}^{n} \geq 0 \) with a sufficiently large number \( M \). Consequently, we have

\[
\bar{O}_{k}^{n} \geq S_{p}^{n} + d_{p}^{n} + t_{p,0}^{n} - L - M(1 - z_{p,0,k}), \quad \forall p \in P,
\]

which implies that \( \bar{O}_{k}^{n} \) satisfies constraints (3m).

J) Let \( \bar{G}_{p}^{n} = \sum_{i \in I} \sum_{k \in K} \bar{x}_{i,p,k} \bar{g}_{i,k}^{n} \) for all \( p \in P \) and \( n \in [N] \). To verify that \( \bar{G}_{p}^{n} \) satisfies constraints (3n), for each \( n \in [N] \), we consider the following two cases.

- Case 1: \( p = 0, q \in P \). Let \( S_{0}^{n} = d_{0}^{n} = 0 \). In this case, for a fixed \( q \in P \), constraints (3n)
are equivalent to
\[ G_q^n \geq S_q^n - S_p^n - d_p^n - t_{p,q}^n - M \left( 1 - \sum_{k \in K} z_{p,q,k} \right) \]
\[ \geq S_q^n - S_0^n - d_0^n - t_{0,q}^n - M \left( 1 - \sum_{k \in K} z_{0,q,k} \right) = S_q^n - t_{0,q}^n - M \left( 1 - \sum_{k \in K} z_{0,q,k} \right). \]

If \( \bar{x}_{1,q,k} = z_{0,q,k} = 0 \) for all \( k \in K \), constraints (3n) are relaxed. Otherwise, if \( \bar{x}_{1,q,k'} = z_{0,q,k'} = 1 \) for some \( k' \in K \), we have \( G_q^n = g_{1,k'}^n \). It follows that
\[ G_q^n = g_{1,k'}^n \geq S_{1,k'} - t_{0,q}^n \quad \text{(since } \sum_{p \in P} \bar{x}_{1,p,k} = \bar{x}_{1,q,k'} = 1) \]
\[ \geq S_q^n - t_{0,q}^n \quad \text{(since } S_q^n = S_{1,k'} \text{ by point } (F)) \]

- Case 2: \((p,q) \in P \times P, p \neq q\). If \( \bar{x}_{t-1,p,k'} = \bar{x}_{t',q,k'} = 1 \) for some \( t' \in [2, |I|] \) and \( k' \in K \), by definition of \( S \) in point (F), we have \( S_p^n = S^n_{t-1,k'} \) and \( S_q^n = S^n_{t',k'} \), and accordingly \( G_p^n = g_{t-1,k'}^n \) and \( G_q^n = g_{t',k'}^n \). Hence by constraints (1n), we have
\[ G_q^n = g_{t',k'}^n \geq S_{t',k'} - S^n_{t-1,k'} - \sum_{p \in P} t_{p,q}^n \bar{x}_{t-1,p,k} - \sum_{(p,q) \in P \times P} t_{p,q}^n \bar{x}_{t',p,q,k'} = S_q^n - S_p^n - d_p^n - t_{p,q}^n. \]

Note that if \( \bar{x}_{i_1,p,k_1} = \bar{x}_{i_2,q,k_2} = 1 \) for some \( \{i_1, i_2\} \in I \) such that \( i_2 > i_1 + 1 \) and \( \{k_1, k_2\} \in K \). In this case, \( \bar{x}_{i-1,p,k} \cdot \bar{x}_{i,q,k} = 0 \) for all \( i \in [2, |I|] \) and \( k \in K \) (i.e., customers \( p \) and \( q \) are not visited consecutively by the same provider). Then we have \( \sum_{k \in K} z_{p,q,k} = 0 \) by the conclusion of point (B), and constraints (3n) are relaxed to \( G_p^n \geq 0 \).

Accordingly, we conclude that \( G_p^n \) satisfies constraints (3n).

Therefore, we conclude that \((z, A, W, S, G, O)\) defined above is a feasible solution of model (Z).

The objective value of this solution equals to
\[ \sum_{k \in K} \sum_{p \in P} \lambda \bar{x}_{0,p,k} + \sum_{n \in [N]} \frac{1}{N} \left[ \lambda \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} t_{p,q}^n \bar{x}_{p,q,k} + \lambda \sum_{p \in P} W_p^n + \lambda \sum_{k \in K} O_k^n + \lambda \sum_{p \in P} G_p^n \right]. \]

Note that by construction (see point (A)), we have
\[ \sum_{k \in K} \sum_{p \in P} \lambda \bar{x}_{0,p,k} = \sum_{k \in K} \sum_{p \in P} \lambda \bar{x}_{1,p,k}, \quad \text{(D.2)} \]
and by the construction in point (B), we have
\[ \lambda \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} t_{p,q}^n \bar{x}_{p,q,k} = \lambda \sum_{k \in K} \left( t_{p,q}^n \sum_{(p,q) \in P \times P} \bar{x}_{p,q,k} + t_{0,q}^n \bar{x}_{0,q,k} + t_{p,0}^n \bar{x}_{p,0,k} \right) \]
\[ = \lambda \sum_{k \in K} \left( \sum_{i=2}^{|I|} \sum_{(p,q) \in P \times P} t_{p,q}^n \bar{x}_{i-1,p,k} \bar{x}_{i,q,k} + \sum_{p \in P} t_{p,0}^n \bar{x}_{1,p,k} + \sum_{q \in P} t_{0,q}^n \bar{x}_{1,q,k} \right). \quad \text{(D.3)} \]

We define \( \bar{z}_{i,p,q,k} = \bar{x}_{i-1,p,k} \bar{x}_{i,q,k} \) for all \( i \in [2, |I|], p \in P, q \in P \) and \( k \in K \). By McCormick
inequalities (constraints (1f)-(1h)), for each $n \in [N]$, equation (D.3) is equivalent to
\[
\lambda^t \sum_{k \in K} \sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} t_{p,q}^n z_{p,q,k} = \lambda^t \sum_{k \in K} \left( \sum_{i = 2}^{[I]} \sum_{(p,q) \in P \times P} t_{p,q,i,p,q,k} + \sum_{p \in P} t_{p,0}^n \bar{x}_{1,p,k} + \sum_{q \in P} t_{0,q}^n \bar{x}_{|I|,p,k} \right).
\] (D.4)

By the construction in point (H), we have for each $n \in [N]
\[
\lambda^w \sum_{p \in P} W_p^n = \lambda^w \sum_{i \in I} \sum_{k \in K} \left( \bar{s}_{i,k}^n - \bar{a}_{i,k} \right) x_{i,p,k} = \lambda^w \sum_{i \in I} \sum_{k \in K} \left( \bar{s}_{i,k}^n - \bar{a}_{i,k} \right).
\] (D.5)

By the construction in point (I), we have for each $n \in [N]
\[
\lambda^o \sum_{k \in K} O_k^n = \lambda^o \sum_{k \in K} \bar{a}_{k}.
\] (D.6)

Finally, by the construction in point (J), we have for each $n \in [N]
\[
\lambda^g \sum_{p \in P} G_p^n = \lambda^g \sum_{i \in I} \sum_{k \in K} \bar{g}_{i,p,k} x_{i,p,k} = \lambda^g \sum_{i \in I} \sum_{k \in K} \bar{g}_{i,p,k}.
\] (D.7)

Combining equations (D.2)–(D.7), we show that the objective function value of model (Z) is equal to the objective function value of model (S).

Conversely, suppose that $(z, A, S, W, G, O)$ is an optimal solution to model (Z), we will construct a feasible solution $(\bar{x}, \bar{z}, \bar{a}, \bar{s}, \bar{g}, \bar{o})$ to model (S) with the same objective value.

A) By constraints (3b)–(3e) and (4a), for a fixed $k \in K$, there exists a unique sequence $\{p_1^k, \ldots, p_{[I]}^k\}$ such that $z_{0,p_1^k,k} = 1$, $z_{p_i^k,p_{i+1}^k,k} = 1$ for all $i \in [1, |I| - 1]$ and $z_{p_{|I|}^k,0,k} = 1$. Then, for each $p_i^k \in P$ and $i \in I$, we define $\bar{x}_{i,p_i^k,k} = 1$. Consequently, $\bigcup_{k \in K} \{p_1^k, p_2^k, \ldots, p_{|I|}^k\}$ forms a partition of $P$, implying that $\bar{x}$ satisfies constraints (1b) and (2c).

B) Using the construction in point (A), we let $\bar{z}_{i,p,q,k} = \bar{x}_{i-1,p,k} \bar{x}_{i,q,k}$, for all $i \in [2, |I|], p \in P, q \in P$ and $k \in K$, for a given $\bar{x}_{i-1,p,k}$ and $\bar{x}_{i,q,k}$. One can easily verify that $\bar{z}$ satisfies constraints (1f)–(1h) by McCormick inequalities.

C) Let $\bar{a}_{i,k} = \sum_{p \in P} A_p \bar{x}_{i,p,k}$ for all $i \in I$ and $k \in K$. For a fixed $(i', k') \in I \times K$, we have $\bar{x}_{i',p',k'} = 1$ (from the construction in point (A)). Then, by constraints (3g), we have $\bar{a}_{i',k'} = \sum_{p \in P} A_p \bar{x}_{i',p,k'} = A_{p_{i'}} \leq L$. (Thus $\bar{a}_{i,k}$ satisfies constraints (2d)).

D) Let $\bar{s}_{i,k}^n = \sum_{p \in P} S^n_p \bar{x}_{i,p,k}$ for all $i \in I, k \in K$ and $n \in [N]$. By the result of point (A), for a fixed $(i', k') \in I \times K$, there exists $p_{i'} \in P$ such that $\bar{x}_{i',p_{i'},k'} = 1$. Since $S$ satisfies constraints (3g) and (3j) and $\bar{a} \leq L$ (as shown in point (C)), for each $n \in [N]$, we have $\bar{s}_{i',k'} = S^n_p \bar{x}_{i',p,k'} = S^n_p \geq A_{p_{i'}} = \sum_{p \in P} A_p \bar{x}_{i',p,k'} = \bar{a}_{i',k'}$, which satisfies constraints (1j). It remains to verify that $\bar{s}_{i,k}^n = \sum_{p \in P} S^n_p \bar{x}_{i,p,k}$ satisfies constraints (1k) and (2f). Consider the following two cases for a fixed $k \in K$ and each $n \in [N]$:

- $i = 1$. By construction of $\bar{x}$ and conclusion of point (A), there must exist $p_1^k \in P$ such
that $z_{0,p_{1}^{k},k} = \bar{x}_{1,p_{1}^{k},k} = 1$. Since $S_{0}^{n} = 0$ (by constraints (3i)) and $d_{0}^{n} = 0$, we have

$$\bar{s}_{1,k}^{n} = \sum_{p \in P} S_{p}^{n} \bar{x}_{1,p,k} = S_{p_{1}^{k}}^{n}$$

$$\geq S_{0}^{n} + d_{0}^{n} + t_{0,p_{1}^{k}}^{n} - M(1 - \sum_{k' \in K} z_{0,p_{1}^{k'},k'}) \quad \text{(since } S_{p_{1}^{k}}^{n} \text{ satisfies constraints (3k)})$$

$$= t_{0,p_{1}^{k}}^{n} = \sum_{p \in P} t_{0,p} \bar{x}_{1,p,k}.$$ 

(since $\bar{x}$ satisfies constraints (2c)).

$- i \in [2, |I|]$. Given $i' \in [2, |I|]$, by definitions of $\bar{x}$ and $\bar{z}$ in points (A) and (B), there must exist $(p_{i' - 1}, p_{i}') \in P \times P$ such that $\bar{x}_{i' - 1,p_{i' - 1},k} = \bar{x}_{i',p_{i}'-1,k}$, $\bar{z}_{i',p_{i}'-1,k} = 1$. By constraints (3k), we have

$$\bar{s}_{i',k}^{n} = \sum_{p \in P} S_{p}^{n} \bar{x}_{i',p,k} = S_{p_{i}'}^{n} \quad \text{(since } S_{p_{1}^{k}}^{n} \text{), Using the fact that}$$

$$S_{0}^{n} = 0,$$

we have

$$\bar{g}_{i,k}^{n} = \sum_{p \in P} G_{p}^{n} \bar{x}_{1,p,k} = G_{p_{1}^{k}}^{n}$$

$$\geq S_{p_{1}^{k}}^{n} - t_{0,p_{1}^{k}}^{n} - M(1 - \sum_{k' \in K} z_{0,p_{1}^{k'},k'}) \quad \text{(since } G_{p_{1}^{k}}^{n} \text{ satisfies constraints (3n)})$$

$$= \bar{s}_{1,k}^{n} - t_{0,p_{1}^{k}}^{n} = \bar{s}_{1,k}^{n} - \sum_{p \in P} t_{0,p} \bar{x}_{1,p,k}.$$ 

(since $\bar{x}_{1,p,k} = \bar{x}_{1,p_{1}^{k},k} = 1$)

$- i \in [2, |I|]$. Given $i' \in [2, |I|]$, by definitions of $\bar{x}$ and $\bar{z}$ in points (A) and (B), there must exist $(p_{i' - 1}, p_{i}') \in P \times P$ such that $\bar{x}_{i' - 1,p_{i' - 1},k} = \bar{x}_{i',p_{i}'-1,k}$, $\bar{z}_{i',p_{i}'-1,k} = 1$. Using the definition of $\bar{s}$ in point (D), we know that $\bar{s}_{i',k}^{n} = S_{p_{i}'}^{n}$ and $\bar{s}_{i'-1,k}^{n} = S_{p_{i}'-1}^{n}$. Then we have

$$\bar{g}_{i',k}^{n} = \sum_{p \in P} G_{p}^{n} \bar{x}_{i',p,k} = G_{p_{i}'}^{n}$$

$$\geq S_{p_{i}'}^{n} - S_{p_{i}'-1}^{n} - d_{p_{i}'}^{n} - t_{0,p_{i}'-1}^{n} \bar{z}_{i',p_{i}'-1,k} - M(1 - \sum_{k' \in K} z_{p_{i}'-1,p_{i}'}^{n},k') \quad \text{(by constraints (3n))}$$

$$= \bar{s}_{i',k}^{n} - \bar{s}_{i'-1,k}^{n} - \sum_{p \in P} d_{p}^{n} \bar{x}_{i'-1,p,k} - \sum_{(\bar{p},\bar{q}) \in P \times P} t_{\bar{p},\bar{q}} \bar{z}_{i',\bar{p},\bar{q},k}. $$

Consequently, we conclude that $\bar{g}$ satisfies constraints (1m) and (1n).

F) Let $\bar{g}_{k}^{n} = O_{k}^{n}, \forall k \in K, n \in [N]$. If $z_{p,0,k} = 0, \forall p \in P \cup \{0\}$ for a fixed $k \in K$, then constraints (3m) are relaxed to $O_{k}^{n} \geq 0$, for each $n \in [N]$. Now, given $\bar{x}$ from point (A), we know that
there must exist $p^k_{|I|} \in P$ such that $z_{p^k_{|I|},0,k} = \bar{x}_{|I|,p^k_{|I|},k} = 1$. Using $\bar{s}$ as defined in point (D), we have
\[
\hat{o}^n_k = O^n_k \geq S^n_{p'} + d^n_{p',0} - L - M(1 - \sum_{k' \in K} z_{p',0,k'}) \quad \text{(since } O^n_k \text{ satisfies constraints (3m)})
\]
\[
\geq \bar{s}_{|I|,k} + \sum_{p \in P} d^n_{p,0}\bar{x}_{|I|,p,k} + \sum_{p \in P} t^n_{p,0}\bar{x}_{|I|,p,k} - L. \quad \text{(since } \sum_{p \in P} \bar{x}_{|I|,p,k} = \bar{x}_{|I|,p^k_{|I|},k} = 1)
\]
Thus, $\hat{o}$ satisfies constraints (2g).

From points (A)–(F), we conclude that $(\bar{x}, \bar{z}, \bar{a}, \bar{s}, \bar{g}, \bar{o})$ is a feasible solution of model (S). The objective value of this solution equals
\[
\sum_{k \in K} \sum_{j \in P} \lambda^f \bar{x}_{1,p,k} + \frac{1}{N} \left[ \lambda^t \sum_{k \in K} \sum_{i=2}^{|I|} \sum_{(p,q) \in P \times P} t^n_{p,q,z_{i,p,q,k}} + \sum_{p \in P} t^n_{p,0} \bar{x}_{1,p,k} + \sum_{q \in P} t^n_{0,q} \bar{x}_{|I|,p,k} \right] + \lambda^n \sum_{i \in I} s_{i,k} - \sum_{i \in I} \bar{a}_{i,k} + \lambda^g \sum_{k \in K} g^n_{k} \right].
\]
Using the logic similar to equations (D.2)–(D.7), we conclude that (D.8) is equivalent to
\[
\sum_{k \in K} \sum_{p \in P} \lambda^f z_{0,p,k} + \frac{1}{N} \left[ \lambda^t \sum_{k \in K} \sum_{p \in P, q \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} W^n_{p,q} + \lambda^n \sum_{k \in K} O^n_k + \lambda^g \sum_{p \in P} G^n_{p} \right].
\]
i.e., the optimal value of model (S) equals the optimal value of model (Z). \hfill \Box

Appendix E. Proof of Theorem 2

We provide proof of Theorem 2, showing that the linear programming relaxation (LPR) of model (S) for the partially used provider case provides a tighter lower bound than the LPR of model (Z). We let LP(S) and LP(Z) respectively represent the optimal objective values of LPR(S) and LPR(Z). Our proof has the following steps. First, we derive a valid upper bound on LP(Z). Second, we derive a lower bound on LP(S). Then, we compare the difference between these bounds.

**STEP 1.** In Theorem 4, we construct a feasible solution for LPR(Z) and use it to establish an upper bound on the optimal objective value of LPR(Z), denoted as LP(Z).

**Theorem 4.** The optimal objective value of LPR(Z), denoted as LP(Z), satisfies
\[
LP(Z) \leq UB_Z = \left[ \lambda^f |P| + \sum_{n \in |N|} \sum_{p \in P} \lambda^n \left( t^n_{0,p} + t^n_{0,0} \right) \right] \frac{d}{V}, \quad \text{(E.1)}
\]
for some large positive constant $V > 0$ (e.g., $V = M$).

**Proof of the Theorem 4.** We construct the following feasible solution to LPR(Z). In this solution we let $V = M = L + (|I| + 1)(t^{\text{max}}_1 + t^{\text{max}}_2) + (|I| - 1)\bar{t}$. Given the number of positions on the serving sequence $|I|$ and the set of providers $K$, we consider a partition of the set of customers $P = \bigcup_{k \in K} \{P_1, P_2, \ldots, P_k\}$ such that $P_k \cap P_{k'} = \emptyset$ and $0 \leq |P_k| \leq |I|$, for all $k \in K, k' \in K : k \neq k'$. Then, we use this partition to construct the following feasible solution $(z, A, S, W, G, O)$ to LPR(Z).
• For any provider $k \in K$ and $(p, q) \in (P \cup \{0\}) \times (P \cup \{0\})$,

$$z_{p,q,k} = \begin{cases} 
1 - |P_k| \frac{d}{M}, & \text{if } p = q = 0 \\
\frac{d}{M}, & \text{if } p = 0, q \in P_k \text{ or } q = 0, p \in P_k \\
1 - \frac{d}{M}, & \text{if } p, q \in P_k : p = q \\
0 & \text{otherwise}
\end{cases} \quad (E.2)$$

• $A_p = 0, \ \forall p \in P; \ S^n_p = 0, \ \forall p \in P, n \in [N]; \ G^n_p = 0, \ \forall p \in P, n \in [N]; \ W^n_p = 0, \ \forall p \in P; n \in [N]$; and $O^n_k = 0, \ \forall k \in K, n \in [N]$.

Let us first show that this solution is a feasible solution for LPR(Z). First, we show that $z_{p,q,k} \in [0,1]$. Since $M \geq |I|\bar{d}$ and $|P_k| \leq |I|$ by construction, then $0 \leq \frac{d}{M} \leq 1$, $0 \leq 1 - \frac{d}{M} \leq 1$, and $0 \leq 1 - |P_k| \frac{d}{M} \leq 1$. It follows that $z_{p,q,k} \in [0,1]$, for all $(p, q) \in P \cup \{0\}, (q, p) \in P \cup \{0\}$ and $k \in K$.

Next, we show that solution $(z, A, S, W, G, O)$ as defined above is a feasible solution to LPR(Z) by verifying that it satisfies all constraints.

A) By construction, for any $p \in P$, there must exist a unique $\bar{k} \in K$ such that $p \in P_k$, i.e., $z_{p,q,k}$ assumes value as defined in (E.2) and $z_{p,q,k} = 0, \ \forall k \neq \bar{k}, q \notin P_k$. Accordingly, we have

$$\sum_{k \in K} \sum_{q \in P_k \cup \{0\}} z_{p,q,k} = \sum_{q \in P_k \cup \{0\}} z_{p,q,k} = \sum_{q \in P_k} z_{p,q,k} + z_{p,0,k} = z_{p,p,k} + z_{p,0,k} = 1 - \frac{d}{M} + \frac{d}{M} = 1.$$ 

Thus, $z$ defined in (E.2) satisfies constraints (3c).

B) We show that $z$ defined in (E.2) satisfies constraints (3c). Similar to the argument in point (A), for any $q \in P$, we can always find a unique $\bar{k} \in K$ such that $q \in P_k$. Thus, if $k \in K \setminus \{\bar{k}\}$, we have $z_{p,q,k} = 0$ for all $(p, q) \in P \setminus \{q\}$, which satisfies constraints (3c). When $k = \bar{k}$, we have

$$\sum_{k \in K} \sum_{p \in P \cup \{0\}} z_{p,q,k} - \sum_{k \in K} \sum_{p \in P \cup \{0\}} z_{q,p,k} = \sum_{p \in P \cup \{0\}} z_{p,q,k} - \sum_{p \in P \cup \{0\}} z_{q,p,k} = (\sum_{p \in P} z_{p,q,k} + z_{0,q,k}) - (\sum_{p \in P} z_{q,p,k} + z_{0,q,k}) = (z_{q,q,k} + z_{0,p,k}) - (z_{q,q,k} + z_{0,q,k}) = (1 - \frac{d}{M} + \frac{d}{M}) - (1 - \frac{d}{M} + \frac{d}{M}) = 0.$$ 

Thus, $z$ defined in (E.2) satisfies constraints (3c).

C) For any provider $k \in K$, there is a set of customers $P_k \subset P$ by the partition of $P$ defined in our construction. If $P_k = \emptyset$, by construction, we have

$$\sum_{p \in P \cup \{0\}} z_{p,0,k} = z_{0,0,k} + \sum_{p \in P_k} z_{p,0,k} = z_{0,0,k} = 1 - 0 \cdot \frac{d}{M} = 1.$$ 

On the other hand, when $P_k \neq \emptyset$, we have $z_{p,q,k} = 0$ for all $(p, q) \in (P \setminus \{P_k\}) \times (P \setminus \{P_k\})$.

Then

$$\sum_{p \in P \cup \{0\}} z_{p,0,k} = z_{0,0,k} + \sum_{p \in P_k} z_{p,0,k} = \left(1 - |P_k| \frac{d}{M}\right) + \sum_{p \in P_k} \frac{d}{M} = \left(1 - |P_k| \frac{d}{M}\right) + |P_k| \frac{d}{M} = 1.$$ 

43
Similarly, we have \( \sum_{p \in P \cup \{0\}} z_{0,p,k} = z_{0,0,k} + \sum_{p \in P_k} z_{0,p,k} = 1 \). Thus, \( z \) defined in (E.2) satisfies constraints (3d) and (3e).

D) For a provider \( k \in K \) and corresponding customer set \( P_k \), if \( P_k = \emptyset \), then following the same argument in point (C), we have \( z_{0,0,k} = 1 \) and it is trivial that constraints (3f) hold. Now, consider the case when \( P_k \neq \emptyset \), we have

\[
\sum_{p \in P \cup \{0\}} \sum_{q \in P \cup \{0\}} z_{p,q,k} = \sum_{p \in P_k} \sum_{q \in P_k} z_{p,q,k} + \sum_{p \in P_k} z_{0,q,k} + \sum_{p \in P_k} z_{p,0,k} + z_{0,0,k} = |P_k| \left( 1 - \frac{\bar{d}}{M} \right) + 2 |P_k| \frac{\bar{d}}{M} + \left( 1 - \frac{|P_k| \bar{d}}{M} \right) = |P_k| + 1 \leq |I| + 1 \quad \text{(since } |P_k| \leq |I| \text{ by construction)}
\]

Thus, \( z \) defined in (E.2) satisfies constraints (3f).

E) Since \( A_p^n = S_p^n = 0 \) for all \( p \in P, n \in [N] \), they satisfy constraints (3g), (3i), (3j), and (3l).

F) To verify that \( S_p^n \) and \( z_{p,q,k} \) satisfy constraints (3k), we need to check if they satisfy

\[
S_p^n + \bar{d}_p^n + t_{p,q}^n - M \left( 1 - \sum_{k \in K} z_{p,q,k} \right) \leq 0, \quad \forall p \in P \cup \{0\}, q \in P
\]

To this end, we consider the following three cases:

- Case 1: \( p = 0 \) and \( q \in P \). In this case, following the conclusion in point (A), we have \( \sum_{k \in K} z_{0,q,k} = \frac{\bar{d}}{M} \) by construction. Then, for any \( q \in P \), we have

\[
S_0^n + \bar{d}_0^n + t_{0,q}^n - M \left( 1 - \frac{\bar{d}}{M} \right) = t_{0,q}^n - M \left( 1 - \frac{\bar{d}}{M} \right) \leq t_1^{\max} - M + \bar{d} \leq 0 \quad \text{(since } M > t_1^{\max} + \bar{d} \text{)}
\]

- Case 2: \( p \in P, q \in P \) and \( p = q \). By construction, we have \( \sum_{k \in K} z_{p,q,k} = \sum_{k \in K} z_{p,p,k} = 1 - \frac{\bar{d}}{M} \). Thus, for any \( p \in P \), we have

\[
S_p^n + \bar{d}_p^n + t_{p,p}^n - M \left( 1 - \sum_{k \in K} z_{p,p,k} \right) = d_p^n - M \frac{\bar{d}}{M} \leq 0 \quad \text{(since } \bar{d} = \max_{n \in [N], p \in P} \{d_p^n\}).
\]

- Case 3: \( p \in P, q \in P \) and \( p \neq q \). In this case \( \sum_{k \in K} z_{p,q,k} = 0 \) by construction. Then, for any \( p \in P \) and \( q \in P \), we have

\[
S_p^n + \bar{d}_p^n + t_{p,q}^n - M = d_p^n + t_{p,q}^n - M \leq 0 \quad \text{(since } M > \max_{n \in [N], p \in P} \{t_{p,q}^n\} + \max_{n \in [N], p \in P} \{d_p^n\}).
\]

Thus, the constructed solution satisfies constraints (3k).

G) We verify that the constructed solution satisfies constraints (3m). First, if \( P_k = \emptyset \) for any \( k \in K \), then \( z_{p,0,k} = 0 \) for all \( p \in P \) by construction. In this case, constraints (3m) trivially hold. Second, consider the case when \( P_k \) is not empty. In this case, we have \( z_{p,0,k} = \frac{\bar{d}}{M} \) for all \( p \in P_k \) by construction. Next, we check if the following inequalities hold.

\[
O_k^n - S_k^n - d_p^n - t_{p,0}^n + L + M(1 - z_{p,0,k}) \geq 0, \quad \forall p \in P \quad \text{(E.3)}
\]
Since $O_k^n = S_p^n = 0, \forall k \in K, p \in P, n \in [N]$, inequalities (E.3) reduce to
\[ L + M - \tilde{d} - d_p^n - t_{p,0}^n \geq 0, \quad \forall p \in P. \]
Inequalities (E.4) hold valid since $M > |I|\hat{d} + t_{1}^{\text{max}} + L = |I| \max_{n \in [N], p \in P} \{d_p^n\} + \max_{n \in [N], p \in P} \{t_{p,0}^n\} + L > \tilde{d} + d_p^n + t_{p,0}^n$. Thus, constraints (3m) are satisfied.

H) To check the feasibility of constraints (3n), we need to show that, for any $p \in P \cup \{0\}, q \in P$ and $n \in [N]$, $G_p^n - S_p^n + d_p^n + t_{p,q}^n + M(1 - \sum_{k \in K} z_{p,q,k}) \geq 0$. Similar to point (G), we consider the following three cases:

- Case 1: $p = 0, q \in P$. In this case, $\sum_{k \in K} z_{0,q,k} = \frac{\hat{d}}{M}$. For any $n \in [N]$ and $q \in P$, we have
  \[ G_p^n - S_p^n + d_p^n + t_{p,q}^n + M(1 - \frac{\hat{d}}{M}) = d_p^n + t_{p,q}^n + M - M \frac{\hat{d}}{M} \geq 0. \]
  The result follows because $G_p^n = S_p^n = 0, \forall p \in P, n \in [N]$ by construction and $M > \tilde{d} + t_{1}^{\text{max}}$.

- Case 2: $p \in P, q \in P$ and $p = q$. In this case, $\sum_{k \in K} z_{p,p,k} = 1 - \frac{\hat{d}}{M}$. Thus, for any $p \in P, n \in [N]$, we have
  \[ G_p^n - S_p^n + d_p^n + t_{p,p}^n + M \frac{\hat{d}}{M} = d_p^n + M \frac{\hat{d}}{M} \geq 0 \] (since $(d_p^n, \hat{d}) \geq 0$).

- Case 3: $p \in P, q \in P$ and $p \neq q$. In this case, $\sum_{k \in K} z_{p,q,k} = 0$. By constraints (3n), for all $p \in P, q \in P$ and $n \in [N]$, we have
  \[ G_q^n - S_p^n + d_p^n + t_{p,q}^n + M = d_p^n + t_{p,q}^n + M \geq 0 \] (since $(d_p^n, t_{p,q}^n, M) \geq 0$).
  Thus, the constructed solution satisfies constraints (3n).

From points (A)–(H), we conclude that the constructed solution $(z, A, S, W, G, O)$ is a feasible solution for LPR(Z). The objective function value $UB_Z$ of this feasible solution is as follows:
\[ UB_Z = \sum_{k \in K} \lambda^f |P_k| \frac{\hat{d}}{V} + \sum_{n \in [N]} \sum_{p \in P} \frac{\lambda^t}{N} (t_{0,p}^n + t_{p,0}^n) \frac{\hat{d}}{V}. \]
Since this is a feasible solution to LPR(Z), the objective value of this solution denoted as $UB_Z$ provides an upper bound on the optimal objective value, $LP(Z)$, of LPR(Z), i.e., $LP(Z) \leq UB_Z$.

Moreover, since $\sum_{k \in K} |P_k| = |P|$, we have
\[ LP(Z) \leq UB_Z = \left[ \lambda^f |P| + \sum_{n \in [N]} \sum_{p \in P} \frac{\lambda^t}{N} (t_{0,p}^n + t_{p,0}^n) \right] \frac{\hat{d}}{V}. \]
This completes the proof of Theorem 4. \qed

STEP 2. In Theorem 5, we derive a lower bound on the optimal objective value $LP(S)$ of LPR(S)

**Theorem 5.** The optimal objective value of the LPR(S), denoted as $LP(S)$, satisfies
\[ LP(S) \geq \frac{|P|}{|I|} (\lambda^f + \lambda^t \cdot t_{1}^{\text{min}}) = LB_S. \]
Proof of the Theorem 5. Let \((x, a, z, s, g, o)\) be an optimal solution to LPR(S). The objective value of this solutions is

\[
LP(S) = \sum_{p \in P} \sum_{k \in K} \gamma^f x_{1,p,k} + \sum_{n \in [N]} \frac{1}{N} \left\{ \lambda^f \left( \sum_{(p,p') \in P \times P} \sum_{i \in I} \sum_{k \in K} t^n_{p,p'} z_{i,p,p',k} + \sum_{p \in P} t^n_{p,0} x_{1,p,k} \right) \right. \\
+ \sum_{p \in P} t^n_{p,0} x_{0,p,k} \right\} + \sum_{k \in K} \sum_{i \in I} \left[ \lambda^u (s^n_{i,k} - a_{i,k}) + \lambda^g g^n_{i,k} \right] + \sum_{k \in K} \lambda^o o^n_k \}
\]

Since decision variables \((x, z, s, a, g, o)\) are non-negative, we have

\[
LP(S) \geq \sum_{p \in P} \sum_{k \in K} \gamma^f x_{1,p,k} + \sum_{n \in [N]} \frac{1}{N} \left[ \sum_{p \in P} \sum_{k \in K} t^n_{p,0} x_{1,p,k} \right] \\
= (\lambda^f + \lambda^t_{1,\text{min}}) \sum_{p \in P} \sum_{k \in K} x_{1,p,k}.
\]

(E.5)

Next, we claim that for any \(k \in K\), there exists some \(p' \in P\) such that

\[
x_{1,p',k} \geq \frac{1}{|I||K|}
\]

(E.6)

We prove this claim by contradiction. Suppose, on the contrary, that \(x_{1,p,k} < 1/|I||K|\) for all \(p \in P\) and \(k \in K\). Summing over the customers set \(P\) we have

\[
\sum_{p \in P} x_{1,p,k} < \frac{1}{|I||K|} = \frac{|P|}{|I||K|}, \quad \forall k \in K.
\]

(E.7)

Form constraints (1e), we know that any feasible solution to LPR(S) should satisfies \(\sum_{p \in P} x_{i,p,k} \leq \sum_{p \in P} x_{1,p,k}\) for all \(i \in [2,|I|]\) and \(k \in K\). Thus, from (E.7), we have \(\sum_{p \in P} x_{i,p,k} < |P|/|I||K|\), for all \(i \in [2,|I|]\) and \(k \in K\). Summing over the position set \(I\) and provider set \(K\), we have

\[
\sum_{i \in I} \sum_{p \in P} \sum_{k \in K} x_{i,p,k} < |P|/|I||K| = |P|.
\]

(E.8)

Note that any feasible solution should also satisfy constraints (1b), i.e., \(\sum_{i \in I} \sum_{k \in K} x_{i,p,k} = 1, \forall p \in P\), which implies that \(\sum_{i \in I} \sum_{p \in P} \sum_{k \in K} x_{i,p,k} = |P|\). But from (E.8), we have \(\sum_{i \in I} \sum_{p \in P} \sum_{k \in K} x_{i,p,k} < |P|\). Hence, we have a contradiction. This complete the proof of our claim. Summing inequality (E.6) over the customer set \(P\) and provider set \(K\), we obtain

\[
\sum_{p \in P} \sum_{k \in K} x_{1,p,k} \geq \frac{|P|}{|I||K|} \cdot |K| = \frac{|P|}{|I|}.
\]

(E.9)

Combining (E.9) with (E.5), we conclude

\[
LP(S) \geq \frac{|P|}{|I|} (\lambda^f + \lambda^t_{1,\text{min}}).
\]

This completes the proof of Theorem 5.

\[\square\]

**STEP 3.** From Theorem 4, we know that the optimal objective value of LPR(Z) is not greater than the constructed upper bound \(UB_Z\) in (E.1), i.e.,

\[
LP(Z) \leq UB_Z = \left[ \lambda^f |P| + \sum_{n \in [N]} \sum_{p \in P} \frac{\lambda^n}{N} \left(t^n_{0,p} + t^n_{p,0}\right) \right] \frac{d}{V} \leq \left[ \lambda^f + \lambda^t \left( t^{\max}_{1} + t^{\max}_{2} \right) \right] \frac{d|P|}{V},
\]

(E.10)
where the last inequality follows from $t_{1}^{\text{max}} = \max_{p \in P, n \in [N]} \{t_{n, p}^{m}\}$ and $t_{2}^{\text{max}} = \max p \in P, n \in [N]\{t_{n, p, 0}^{m}\}$. On the other hand, from Theorem 5, we know that the optimal objective value of $\text{LPR}(S)$ satisfies $LP(S) \geq LB_S = \left|P\right| \left(\lambda f + \lambda t_{\min, 1}\right)$. The difference between the lower bound on $LP(S)$ ($LB_S$) and the upper bound of $\text{LPR}(Z)$ defined in (E.10) is as follows

$$V - \left|I\right| \bar{d} \left|P\right| \lambda f + \left|V_{l_{1}^{\text{min}}}^{\text{max}} \left|I\right| \bar{d} (t_{1}^{\text{max}} + t_{2}^{\text{max}}) \right|P\right| \lambda f$$

(E.11)

Since $V > (\bar{d} |I| + 1)(t_{1}^{\text{max}} + t_{2}^{\text{max}})$, the first and second terms in (E.11) are greater than zero, i.e., the difference between the lower bound on $LP(S)$ and the upper bound on $LP(Z)$ is positive. This indicates that $\text{LPR}(S)$ is tighter than $\text{LPR}(Z)$. Indeed, the numerical results in Sections 7.2.1 and 7.2.2 show that $\text{LPR}(S)$ is strictly tighter than $\text{LPR}(Z)$.

Appendix F. Sample average approximation and sample size

We use the SAA method with Monte Carlo Optimization (MCO) procedure to decide the sample size for the SAA model. We refer to Kenyon and Morton (2003) and Kleywegt et al. (2002) for details of the algorithm. We initialize the MCO procedure with sample size $N$, simulation sample size $N'$, and the number of replicates $M$. In each replicate $m \in [M]$, we first solve the SAA problem with sample size $N$, obtain the optimal solution $\hat{x}^{m}_{N}$, and the optimal objective value $v^{m}_{N}$. Second, we solve the second-stage problem with $\hat{x}^{m}_{N}$ and $N'$ scenarios to compute $v^{m}_{N'}$. We repeat these steps $M$ times, each time with new $N$ and $N'$ scenarios of service and travel time sampled from their distributions. Finally, we compute the average of the SAA objective value $\bar{v}_{N} = \frac{1}{M} \sum_{k \in K} v^{m}_{N}$ and the simulated objective values $\bar{v}_{N'} = \frac{1}{M} \sum_{k \in K} v^{m}_{N'}$. As detailed in Kenyon and Morton (2003) and Kleywegt et al. (2002), $\bar{v}_{N}$ and $\bar{v}_{N'}$ respectively represent statistical lower and upper bounds on the optimal value of the HFASP. Thus, we estimate the Approximate Optimality Index $\text{AOI} = \frac{v^{m}_{N'} - v^{m}_{N}}{v^{m}_{N'}}$.

We implement the algorithm using model (S) for partially used providers with the problem instance $|I| = 6, |P| = 24, |K| = 5$ under the cost structure defined in Section 7.1. We run the experiment with the sample size $N$ ranging from 1 to 100. For each value of $N$, we repeat the algorithm ten times ($M = 10$) and choose the Monte Carlo simulation sample size $N' = 10000$. We present SAA objective value $v_{N}$, objective function value of simulation $v_{N'}$, $\text{AOI}$, and their 95% Confidence Interval (95%CI) in Table F1. This table shows that $\text{AOI}$ with $N = 50$ equals 0.01%. In addition, the 95%CI of $\bar{v}_{N=50}$ and $\bar{v}_{N'}$ are very tight. These results qualify $v_{N=50}$ as a tight estimator of the optimal value.

Appendix G. Additional computational results for model (S) of fully used provider

We provide additional computational time results of model (S) for fully used providers. We present the results in two parts. In the first part, we analyze the solution times of model (S) and compare them with that of model (Z) under 100 scenarios. In the second part, we present solution times of
Table F1: $v_N$, $v_{N'}$, and AOI of the partially used model with $|I| = 6$, $|P| = 24$ and $|K| = 5$

| $N$ | $\bar{v}_N$  | $\bar{v}_{N'}$ | $\text{AOI(\%)}$ | 95%CI $\bar{v}_N$ | 95%CI $\bar{v}_{N'}$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4052.98</td>
<td>4186.69</td>
<td>3.19</td>
<td>[4050,4056]</td>
<td>[4171,4203]</td>
</tr>
<tr>
<td>5</td>
<td>4053.23</td>
<td>4061.81</td>
<td>0.21</td>
<td>[4052,4054]</td>
<td>[4059,4065]</td>
</tr>
<tr>
<td>10</td>
<td>4054.45</td>
<td>4057.24</td>
<td>0.07</td>
<td>[4054,4055]</td>
<td>[4056,4058]</td>
</tr>
<tr>
<td>20</td>
<td>4054.57</td>
<td>4056.65</td>
<td>0.05</td>
<td>[4054,4055]</td>
<td>[4056,4057]</td>
</tr>
<tr>
<td>30</td>
<td>4054.49</td>
<td>4058.98</td>
<td>0.11</td>
<td>[4054,4055]</td>
<td>[4058,4060]</td>
</tr>
<tr>
<td>40</td>
<td>4054.50</td>
<td>4055.81</td>
<td>0.03</td>
<td>[4054,4055]</td>
<td>[4056,4056]</td>
</tr>
<tr>
<td>50</td>
<td>4055.06</td>
<td>4055.42</td>
<td>0.01</td>
<td>[4055,4056]</td>
<td>[4055,4056]</td>
</tr>
<tr>
<td>60</td>
<td>4054.73</td>
<td>4055.58</td>
<td>0.02</td>
<td>[4054,4055]</td>
<td>[4055,4056]</td>
</tr>
<tr>
<td>70</td>
<td>4055.03</td>
<td>4056.34</td>
<td>0.03</td>
<td>[4055,4055]</td>
<td>[4056,4057]</td>
</tr>
<tr>
<td>80</td>
<td>4054.70</td>
<td>4054.47</td>
<td>0.01</td>
<td>[4054,4055]</td>
<td>[4054,4055]</td>
</tr>
<tr>
<td>90</td>
<td>4054.59</td>
<td>4055.20</td>
<td>0.02</td>
<td>[4054,4055]</td>
<td>[4055,4055]</td>
</tr>
<tr>
<td>100</td>
<td>4054.82</td>
<td>4054.74</td>
<td>0.01</td>
<td>[4055,4055]</td>
<td>[4055,4055]</td>
</tr>
</tbody>
</table>

Table G1: Solution time (in seconds) of model (S) for fully used providers with 100 scenarios

| model (S) | $|I| = 6$ |  |  | $|I| = 8$ |  |  |
|-----------|---------|---|---|---------|---|---|
|           | $|P|$    | Min | Avg | Max    | $|P|$    | Min | Avg | Max    |
|           | 24      | 4.8 | 5.4 | 5.9    | 24      | 4.8 | 5.3 | 5.1    |
|           | 30      | 11.6| 13.9| 19.5   | 32      | 20.3| 22.9| 25.4   |
|           | 36      | 28.7| 31.9| 39.7   | 40      | 47.3| 67.0| 114.2  |
|           | 42      | 88.6| 390.7| 682.6  | 48      | 793.0| 1142.3| 1656.3 |
|           | 48      | 248.9| 658.4| 1240.4 | 56      | 2388.3| 2860.7| 3500.3 |
|           | 54      | 1339.3| 2116.3| 2817.1 | 64      | -    | -    | -      |

When $|I| = 8, |P| \geq 32$, we set the relative MIP gap to 0.04.

We start with reporting solution times of model (S) using $N = 100$ scenarios in Table G1. First, we observe that solution time increases when $N$ increases from 50 to 100 scenarios. In fact, the average solution time of model (S) with $|I| = 6$ ranges from 5.4 seconds ($|P| = 24$) to 35 minutes ($|P| = 54$), with $|I| = 8$ ranges from 7.1 seconds ($|P| = 24$) to 47 minutes ($|P| = 56$). In contrast, using model (Z), we were only able to solve instances with $|I| = 6, |P| \leq 18$ or $|I| = 8, |P| \leq 16$.

We compare the ratios of solution times of model (Z) and (S) in Table G2. Clearly, solution times of model (Z) are longer than model (S). Next, we present numerical results under a different cost structure (hereafter denoted as cost structure 2). Specifically, as in Yu et al. (2021), we set the unit overtime cost $\lambda^o = 10$, and unit waiting cost $\lambda^w = 2$. The other elements in the cost structure are the same. We generate unit travel cost $\lambda^t$ from $U[0.1, 0.5]$ (Zhan and Wan, 2018) and set the fixed cost of hiring one provider $\lambda^f$ to 1000 based on real-world applications. In Table G3, we present the solution time of model (S) under cost structure 2 and 50 scenarios. We observe that using model (S), the average solution time ranges from 3.4 seconds ($|P| = 24$) to 13 minutes ($|P| = 60$).

Finally, we compare the solution time of two models with cost structure 2 and present the ratio of solution times of models (S) and (Z) in Table G4. Observe that, model (Z) takes a longer time to solve all instances than model (S). For those instances that model (Z) fails to solve, it terminates with the average relMIP around 6%.
Table G2: Ratios of solution times of models (Z) and (S) on the SAAs solved by both with $N = 100$ (fully used)

| $|P|$ | (Z) sol.time $|I| = 6$ | (S) sol.time $|I| = 6$ | $|I| = 8$ |
|------|----------------|----------------|----------------|
| 6    | 1.6            | 2.4            | 3.9            |
| 12   | 2.0            | 3.2            | 5.5            |
| 18   | 148.7          | 187.7          | 305.0          |

Table G3: Solution time (in seconds) of model (S) for fully used provider case with cost structure 2

| $|P|$ | $|I| = 6$ | $|I| = 8$ |
|------|----------|----------|
| 24   | 3.2      | 3.4      |
| 30   | 7.9      | 9.0      | 11.6     |
| 36   | 18.0     | 21.5     | 28.3     |
| 42   | 26.5     | 142.2    | 192.4    |
| 48   | 155.7    | 261.6    | 392.9    |
| 54   | 245.0    | 861.4    | 1993.4   |
| 60   | 110.5    | 777.6    | 2221.9   |

Table G4: Ratios of solution times (in seconds) of models (Z) and (S) on the SAAs solved by both with cost structure 2. Results are for fully used models

| $|P|$ | (Z) sol.time $|I| = 6$ | (S) sol.time $|I| = 6$ | $|I| = 8$ |
|------|----------------|----------------|----------------|
| 6    | 2.0            | 2.5            | 3.1            |
| 12   | 1.1            | 2.0            | 2.2            |
| 18   | 250.6          | 335.0          | 447.8          |

Appendix H. Additional computational results for model (S) of partially used provider

In this section, we provide additional computational time results of model (S) for partially used providers with 100 scenarios and cost structure 2. The experiment settings are same as what we discussed in Section 7.1, and details about cost structure 2 are described in Appendix G.

First, we report the Min, Avg and Max solution time of generated instances with $N = 100$ in Table H1. We observe that the solution time of model (S) with $|I| = 6$ varies from 7.8 seconds ($|P| = 24$) to 53 minutes ($|P| = 54$), and with $|I| = 8$ varies from 17.3 seconds ($|P| = 24$) to 20 minutes ($|P| = 42$). In contrast, model (Z) was able to solve only instances with $|I| = \{6, 8\}$, $|P| \leq 8$.

We compare the ratios of solution times of model (Z) and (S) in Table H2. It is clear that solution time of model (Z) is longer than that of model (S). For those instances that model (Z) failed to solve, it terminated with the average relMIP gap around 74% ($|I| = 6$) and 100% ($|I| = 8$).

Finally, we present results with cost structure 2 in Table H3. We observe that the average solution time of model (S) with $|I| = 6$ ranges from 7.1 ($|P| = 24$) seconds to 9.5 minutes ($|P| = 62$). In contrast, model (Z) can only solve instances with $|I| \subset \{6, 8\}$, $|P| \leq 8$. The average relMIP at termination is 100%. We present the comparison of solution times between models (S) and (Z) in
Table H1: Solution time (in seconds) of model (S) for partially used providers with 100 scenarios

| model (S) | $|I| = 6$ | $|I| = 8$ |
|-----------|-----------|-----------|
| $|P|$ | Min | Avg | Max | Min | Avg | Max |
| 24        | 7.8      | 9.6      | 10.8 | 16.6 | 17.3 | 22.3 |
| 30        | 27.6     | 31.1     | 34.0 | 51.8 | 65.8 | 80.9 |
| 36        | 60.9     | 64.7     | 69.8 | 331.3| 443.1| 525.0|
| 40        | 106.7    | 121.6    | 134.3| 1127.3| 2419.8| 3089.4|
| 42        | 121.5    | 146.0    | 178.5| 914.2| 1218.5| 1532.8|
| 48        | 965.3    | 1361.3   | 1590.5| -    | -    | -    |
| 50        | 991.3    | 1661.9   | 2334.3| -    | -    | -    |
| 54        | 1166.0   | 2520.7   | 3170.7| -    | -    | -    |

Table H2: Ratios of solution time of models (Z) and (S) on the SAAs solved by both with $N = 100$ (partially used)

| (Z) sol.time | (S) sol.time | $|I| = 6$ | $|I| = 8$ |
|--------------|--------------|-----------|-----------|
| $|P|$ | Min | Avg | Max | Min | Avg | Max |
| 6           | 2.2 | 5.1 | 7.8 | 1.1 | 2.9 | 3.9 |
| 8           | 86.2 | 95.8 | 100.1 | 9.2 | 21.0 | 54.0 |

Table H3: Solution time (in seconds) of model (S) for partially used providers with cost structure 2

| $|I| = 6$ ($\lambda_w, \lambda_o, \lambda_g) = (2,10,0)$ | $|P|$ | Min | Avg | Max |
|--------------------------------------------------|-------|-----|-----|-----|
| 24                                               | 6.1  | 7.1 | 9.0 |
| 30                                               | 13.5 | 14.5| 15.4|
| 36                                               | 16.2 | 16.8| 18.3|
| 40                                               | 23.7 | 25.5| 27.7|
| 42                                               | 28.3 | 29.0| 30.1|
| 48                                               | 43.4 | 46.6| 52.3|
| 50                                               | 60.7 | 132.4| 340.1|
| 54                                               | 81.4 | 85.6| 96.0|
| 58                                               | 115.7| 197.2| 311.2|
| 60                                               | 119.7| 136.4| 150.0|
| 62                                               | 163.6| 574.9| 1599.9|

Table H4: Ratios of solution time of models (S) and (Z) on the SAAs solved by both with cost structure 2. Results are for partially used models

| (S) sol.time | (Z) sol.time | $|I| = 6$ ($\lambda_w, \lambda_o, \lambda_g) = (2,10,0)$ | $|I| = 8$ |
|--------------|--------------|--------------------------------------------------|-----------|
| $|P|$ | Min | Ave | Max | Min | Ave | Max |
| 6           | 1.5 | 1.7 | 1.8 | 0.8 | 1.0 | 1.3 |
| 8           | 69.8 | 73.6 | 80.5 | 3.3 | 3.5 | 4.0 |

Table H4. It is clear that model (Z) takes a longer time to solve all instances than model (S).
Appendix I. Symmetry breaking constraints

Suppose there are three homogeneous providers $K = \{1, 2, 3\}$, i.e., they share the same hiring cost $\lambda$ and have same service time distribution. Then, solutions $(\sum_{p \in P} x_{1,p,1} = 1, \sum_{p \in P} x_{1,p,2}), (\sum_{p \in P} x_{1,p,1} = 1, \sum_{p \in P} x_{1,p,3} = 1), \text{ and } (\sum_{p \in P} x_{1,p,2} = 1, \sum_{p \in P} x_{1,p,3} = 1)$ are equivalent (i.e., yield the same objective) in the sense that they all permit hiring 2 out of 3 providers. To prevent wasting time exploring such equivalent solutions, we assume that providers are numbered sequentially and add constraints (I.1) to model (S). Similarly, we add constraints (I.2) to model (Z) to enforce that provider $k$ is hired before provider $k + 1$.

$$\sum_{p \in P} x_{1,p,k} \geq \sum_{p \in P} x_{1,p,k+1}, \quad \forall k \in [1, |K| - 1]. \quad \text{(I.1)}$$

$$\sum_{p \in P} z_{0,p,k} \geq \sum_{p \in P} z_{0,p,k+1}, \quad \forall k \in [1, |K| - 1]. \quad \text{(I.2)}$$
Appendix J. Details of Lehigh County Instances

Table J1: The number of customers in each city/township of Lehigh Valley Instance

<table>
<thead>
<tr>
<th>City</th>
<th>Pop</th>
<th>Pop%</th>
<th>L-50</th>
<th>L-100</th>
<th>City</th>
<th>Pop</th>
<th>Pop%</th>
<th>L-50</th>
<th>L-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alburtis</td>
<td>2596</td>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>Bethlehem</td>
<td>25868</td>
<td>6.6</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Coplay</td>
<td>3348</td>
<td>0.9</td>
<td>1</td>
<td>1</td>
<td>Hanover</td>
<td>11783</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Heidelberg</td>
<td>3324</td>
<td>0.9</td>
<td>1</td>
<td>1</td>
<td>Lower Macungie</td>
<td>32426</td>
<td>8.3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Lower Milford</td>
<td>3861</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Lynn</td>
<td>4232</td>
<td>1.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Lowhill</td>
<td>2292</td>
<td>0.6</td>
<td>1</td>
<td>1</td>
<td>Salisbury</td>
<td>13621</td>
<td>3.5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>North Whitehall</td>
<td>15655</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>South Whitehall</td>
<td>21080</td>
<td>5.4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Weisenberg</td>
<td>4976</td>
<td>1.3</td>
<td>1</td>
<td>1</td>
<td>Upper Macungie</td>
<td>26377</td>
<td>6.8</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Catasauqua</td>
<td>6518</td>
<td>1.7</td>
<td>1</td>
<td>2</td>
<td>Upper Milford</td>
<td>7777</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Coopersburg</td>
<td>2447</td>
<td>0.6</td>
<td>1</td>
<td>1</td>
<td>Upper Saucon</td>
<td>16973</td>
<td>4.3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Emmaus</td>
<td>11652</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>Washington</td>
<td>6551</td>
<td>1.7</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Fountain Hill</td>
<td>4832</td>
<td>1.2</td>
<td>1</td>
<td>1</td>
<td>Whitehall</td>
<td>29173</td>
<td>7.5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Macungie</td>
<td>3257</td>
<td>0.8</td>
<td>1</td>
<td>1</td>
<td>Allentown</td>
<td>125845</td>
<td>32.2</td>
<td>14</td>
<td>32</td>
</tr>
<tr>
<td>Slatington</td>
<td>4283</td>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure J.1: Location of customers in L-50 (left) and L-100 (right)