The Home Service Assignment, Routing, and Appointment Scheduling (H-SARA) problem with Uncertainties*

Syu-Ning Johnn[†] Yiran Zhu[†] Andrés Miniguano-Trujillo[†‡§]
Akshay Gupte[†]

August 7, 2021

Abstract

The Home Service Assignment, Routing, and Appointment scheduling (H-SARA) problem integrates the strategical fleet-sizing, tactical assignment, operational vehicle routing and scheduling subproblems at different decision levels, with a single period planning horizon and uncertainty (stochasticity) from the service duration, travel time, and customer cancellation rate. We propose a two-stage stochastic mixed-integer linear programming model for the H-SARA problem. Additionally, a reduced deterministic version is introduced which allows to solve small-scale instances to optimality with two acceleration approaches. For larger instances, we develop a tailored two-stage decision support system that provides high-quality and in-time solutions based on information revealed at different stages. Our solution method aims to reduce various costs under stochasticity, create reasonable routes with balanced workload and team-based customer service zones, and increase customer satisfaction by introducing a two-stage appointment times update at different times before the actual service. Our two-stage heuristic is competitive to CPLEX's exact solution methods in providing time and cost-effective decisions and can update previously-made decisions based on an increased level of information. Results show that our two-stage heuristic is able to tackle reasonable-size instances and provides good-quality solutions using less time compared to the deterministic and stochastic models on the same set of simulated instances.

Keywords. Stochastic Mixed Integer Programming · Vehicle Routing Problem · A Priori Optimisation · Adaptive Large Neighbourhood Search Heuristics · Home Health Care

AMS 2020 subject classification. 90B15, 90C11, 90C59

1 Background

The Home Service Assignment, Routing, and Appointment scheduling (H-SARA) problem, which was presented for the 13th AIMMS-MOPTA Optimization Modeling Competition [SC21], consists of (1) a *fleet-sizing problem* that determines the number of service teams, (2) an *assignment problem* between service teams and customers, (3) a *vehicle routing problem* that sequences customer visits, and finally (4) a *scheduling problem* that assigns appointment time-slots to all

*This paper is based on the authors' submission to the 13th AIMMS-MOPTA Optimization Modeling Competition at which they won the First prize [SC21].

[†]School of Mathematics, University of Edinburgh, UK.
[‡]Department of Mathematics, Heriot-Watt University, UK.
[§]Maxwell Institute for Mathematical Sciences, UK.

customers with service demand. The H-SARA problem relates to a set of widely studied problems in both academia and industry. First is the *Vehicle Routing Problem* (VRP) which is a generalisation of the well-known *Travelling Salesman Problem* (TSP). The earliest mathematical work on the VRP dates back to Dantzig and Ramser [DR59] on modelling a fleet of capacitated homogeneous trucks to replenish a group of gas stations departing from a central hub. For a typical VRP, the main aim is to determine a set of minimum-distanced tours visiting all the locations starting and ending at a depot, meanwhile satisfying a list of general limitations including space and time capacity, time windows, maximum vehicle travel time, and traversal distance. With numerous applications in logistics, transportation and general distribution management, the VRP has been studied widely in the past few decades and has been extended with several variants and applications [Lap09; Lin+14; BRV16].

Scheduling usually refers to the chronological allocation of tasks to workers such that the list of tasks (components) are accomplished with the shortest amount of time and minimal time clashes. In the H-SARA problem, an appointment time slot is assigned to all customers with service demand. Equivalently, from a service provider's perspective, each customer visit is scheduled as part of a service team's timetable in sequential order. The *Vehicle Routing Problem with Time Windows* (VRP-TW) is a VRP-variant stressing that vehicle arrival and/or departure times must satisfy additional customer availability requirements. We identify the difference of *scheduling* from the VRP-TW as the pro-activeness from the decision-makers: the visiting sequence is the result of initial routing criteria instead of customer-imposed time requirements. Another related problem is the *Home Health Care Routing and Scheduling Problem* (HHC-RSP) in the context of home service [Cis+17; FH17].

In reality, one or multiple elements of the classical VRP is often expected to be uncertain due to the limited availability of information. Common uncertainties include customer presence, traversal times, and service duration. These can be modelled as stochastic random variables, giving rise to the *Stochastic Vehicle Routing Problem* (SVRP) and its variants [OAW18]. The SVRP is usually solved by applying (two-stage) stochastic programming techniques [Sah04; KS17; Tor+19]. *A priori* optimisation [BJO90] works on real-world applications in which randomness is a major concern. It applies a two-stage strategy: in the first stage, an initial solution is created before the parameters are revealed in the second stage. It means that first-stage decisions should possess sufficient flexibility for the second-stage recourse actions.

The idea of *a priori* optimisation can be easily found in reality. Many international shippers (e.g., DPD [DPD20], Royal Mail [Daw19]) have now adapted to similar concepts in their last-mile deliveries: they first assign an estimated time slot to all customers based on the pre-collected information, then re-assign a narrower time slot on the actual day of delivery when more information is known (e.g., customer delivery sequence, cancellations). This multi-stage approach also suits the real-life circumstances in the home healthcare service industry, where last-minute service cancellations, i.e. customers cancelling their requests after being given an appointment time, are allowed. Home service statistics show that the average daily visits per service team in the U.S is around 6, and that driving time typically accounts for 18% to 26% of total working time [HA14], which indicate how a single cancellation can considerably change the timescale for the following visits and the necessity of a robust service planning system. Several works on home service-related research implement this multi-stage approach [FH15; May+15; Rod+15; Shi+18; RRV20].

This multi(two)-stage strategy has became more common in modern-day operational models, evidencing the need to avoid unnecessary visits due to customer non-presence, as well as to support customer flexibility and increase customer satisfaction. In contrast, cancellations have definitely complicated the decision-maker's decisions at each stage.

There are existing works in the literature that deal with travel times, service times, or customer uncertainties in the context of *Home Health Care* (HHC). Readers are referred to [GV13] for reviews of relevant models and methods in HHC. Particularly, [LM11; YLJ15; ZW18; ZWW21] consider randomness in service times. [Shi+18; LYJ19] consider travel and service times uncertainties. [Cap+18] considers customers who request service cancellation, and [Han+17] considers random customer behaviours in attended home delivery. Yet, to our best knowledge, there is no research that integrates all three types of uncertainties with the four decisions in the context of HHC.

The main contribution of this work is the treatment of an H-SARA problem integrating four decisions levels: strategical/tactical level fleet-sizing, tactical level assignment, operational level routing, and operational scheduling. Travel times, service duration, and demand levels are considered jointly as uncertain quantities, which to our best knowledge has not been investigated in the literature before. The developed two-stage heuristic takes parameter evolution into account, thus allowing decision-making based on imperfect information before the actual customer demand is revealed.

2 Stochastic Mixed Integer Programming Formulation

Let a service area be represented by a directed graph $G := (V, A)$. Here the node set V encloses the customer set $[\![1, n]\!] := \{1, \ldots, n\}$, a single depot $\{0\}$, and its duplicate $\{n+1\}$. The arc set consists of all the arcs linking each pair of customers, as well as a single link from the depot to each customer and another from each customer back to the duplicated depot, all with the shortest distance; namely $A := \{(i, j) : i \neq j, \forall i, j \in [\![1, n]\!]\} \cup \{(0, j) : \forall j \in [\![1, n]\!]\} \cup \{(i, n+1) : \forall i \in [\![1, n]\!]\}$. The service for all n customers of known geographical location is provided by a group of no more than m homogeneous service teams, each of which makes a single trip starting from and returning to the depot. We aim to partition the set of customers into the minimum number of groups, each visited exactly once by an individual service team in an explicit visiting sequence, and to determine customer appointment time-slots prior to the actual visits. The solution should satisfy time and capacity constraints given by the customer and the service team. Should service cancellation occurs, the service team will skip the cancelled customer and travel directly to the next scheduled customer instead of visiting the cancelled node.

We apply *a priori* optimisation, where a set of *a priori* vehicle routes is first planned in the presence of estimated expected traversal and service times. The precise duration of each tour becomes available only after the actual travel and service times are revealed in the second stage. Consequently, there is always an inevitable chance of a solution "failing" under the stochasticity setting, forcing the decision-makers to develop relevant recourse policies specifying the actions to repair a failed (infeasible) solution.

An extension beyond the consideration of stochastic travel and service times is the stochastic customer behaviour (customer presence). An option provided by Sörensen and Sevaux [SS09] is to first include all customers in the routes, then remove customer set $\mathbb{I} \in I$ who cancel their service requirements on short notice. This gives a conservative or risk-averse approach for the decision-makers since the routes are feasible for any customer set realisations, provided that the traversal and service times are feasible.

We introduce the stochastic *H-SARA* model with a set Ξ consisting of different scenarios ξ, each associated with a realisation of the travel and service durations. We impose a stochastic traversal duration matrix $T = \tau_{i,j}^{\xi}$ under scenario ξ for any arc $(i, j) \in A$, and a stochastic service duration vector $S = s_i^{\xi}$ for customer i under scenario ξ. The Euclidean distance from i

to j is labelled $d_{i,j}$. In this formulation, symmetry of τ and d is not required, capturing possible discrepancies in the underlying road network; i.e., city topography and street layout. The cost of hiring a team $i \in [\![1, m]\!]$ is taken as a constant f_m. The maximum allowed working time is given by $L \geqslant 0$. Working times are expected to be allocated in the interval $[0, L]$, yet we anticipate possible overtime occurring in the interval $(L, L+\theta]$ with $\theta > 0$. Finally, let c_{wait}, c_{idle}, and c_{over} be fixed non-negative waiting, idling, and overtime costs, respectively. The appointment time of customer i is denoted by t_i and we assume the appointment time window is $[t_i - W, t_i + W]$ with a fixed width $2W$. The arrival of a service team before the appointment time window leads to team idling, whereas a later arrival leads to the customer waiting. Any additional time beyond the maximum working time L results in overtime cost.

For the decision variables, we let $x_{i,j}$ be a binary variable which takes the value of one if the arc $(i, j) \in A$ is traversed by a service team, otherwise it takes the value of zero. We use a continuous variable $0 \leqslant a_i \leqslant L$ for the team's arrival time at customer $i \in [\![1, n]\!]$. Likewise, w_i and h_i are non-negative real-valued variables for the customer's waiting time, and service team's idle time at customer $i \in [\![1, n]\!]$, respectively. g_i is also a real-valued variable measuring the overtime of a service team, registered at their arrival at the depot when returning from customer $i \in [\![1, n]\!]$. Finally, since an actual arrival time under the stochastic setting could be different from a customer's appointment time, we have differentiated an appointment time (scheduled service start time) decision variable t_i for each customer $i \in [\![1, n]\!]$. A summary of the symbols can be found in Table 1.

We have traversal variables $x_{i,j}$ (also fleetsize) and appointment time decision variables t_i as our first-stage decisions, whereas team idle time h_i, overtime g_i, and customer waiting time w_i as our second-stage decisions. The first and second stage formulations for the stochastic *H-SARA* model are as follows:

$$
\begin{aligned}
(1^{st}\text{ Stage}) \qquad \min_{x} \quad & f_m \sum_{i \in [\![1,n]\!]} x_{i,n+1} + \sum_{(i,j) \in A} d_{i,j} x_{i,j} + \mathbb{E}\left[Q(x, \xi)\right] && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{i \in [\![1,n]\!]} x_{0,i} \leqslant m, && && \text{(1b)}\\
& \sum_{i \in [\![0,n]\!]} x_{i,j} = 1 && \forall j \in [\![1, n]\!], j \neq i && \text{(1c)}\\
& \sum_{i \in [\![0,n]\!]} x_{i,j} = \sum_{i \in [\![1,n+1]\!]} x_{j,i} && \forall j \in [\![1, n]\!], j \neq i && \text{(1d)}\\
& \sum_{i \in [\![1,n]\!]} x_{0,i} = \sum_{i \in [\![1,n]\!]} x_{i,n+1}, && && \text{(1e)}\\
& x_{i,j} \in \{0, 1\} && \forall (i, j) \in A. && \text{(1f)}
\end{aligned}
$$

where $\mathbb{E}\left[Q(x, \xi)\right] = \sum_{\xi \in \Xi} q^{\xi} Q(x, \xi)$ for any x satisfying the above equations, and any $\xi \in \Xi$ associated with probability q^{ξ}. The objective function (1a) minimises the total traversal costs, team hiring costs, and expected idling, waiting, overtime costs under all scenarios. Constraint (1b) states that there are a limited number of homogeneous service teams departing from the depot $\{0\}$, limited by m. (1c) require that each customer must be visited once and only once by a service team. The flow conservation constraints (1d) require that a team travelling to any customer node must leave the node afterwards. This is complemented with (1e), which stresses that the number of teams leaving the depot must equal the number that returns. (1f) are the

Table 1: Table of symbols

Symbol	Definition	Description
i, j	Node index	Non-negative integer
M	Number of service teams	Non-negative integer
N	Number of customers	Non-negative integer
$d_{i,j}$	Travel (Euclidean) distance between i and j	Non-negative number
$v_{i,j}$	Traversal speed from node i to j	Positive number
$\tau_{i,j}$	Traversal time from node i to j	Positive number
$\hat{v_{i,j}}$	Expected travelling speed from node i to j	Positive number
$\hat{\tau_{i,j}}$	Expected traversal time from node i to j	Positive number
s_i	Mean service time at customer i	Positive number
p_i	cancellation probability of customer i	Non-negative number
λ	Unit travel time cost (per hour)	Positive number
f_m	Cost for hiring a team	Positive number
c_{wait}	Waiting cost (per unit time)	Non-negative number
c_{idle}	Idle cost (per unit time)	Non-negative number
c_{over}	Overtime cost (per unit time)	Non-negative number
L	Maximum allowed working time	Positive number
L_δ	Safety time at the end of each tour	Positive number
θ	Maximum overtime length	Non negative number
W	A half width of time window	Non-negative number
t_i	Appointment time of customer i	Continuous variable
a_i	Service team's arrival time at customer i	Continuous variable
w_i	Waiting time for customer i	Continuous variable
h_i	Idle time of service team at customer i	Continuous variable
g_i	Overtime length returning to depot from customer i	Continuous variable
$x_{i,j}$	Determines if arc (i, j) is traversed by a team	Integer (binary) variable
ξ	scenario with travel and service times realisations	samples
q_ξ	scenario ξ's associated probability	non-negative number
Ξ	finite set of scenarios $[\xi]$	samples

domain constraints.

$$
\begin{aligned}
(2^{nd}\text{ Stage}) \quad Q(x,\xi) = \min_{w,h,g} \quad & c_{\text{wait}} \sum_{i\in[\![1,n]\!]} w_i^\xi + c_{\text{idle}} \sum_{i\in[\![1,n]\!]} h_i^\xi + c_{over} \sum_{i\in[\![1,n]\!]} g_i^\xi && && \text{(1g)}\\
\text{s.t.} \quad & a_i^\xi + h_i^\xi + s_i^\xi + \tau_{i,j}^\xi \leqslant a_j^\xi + M(1-x_{i,j}) && \forall (i,j)\in A, && \text{(1h)}\\
& a_i^\xi + h_i^\xi + s_i^\xi + \tau_{i,j}^\xi \geqslant a_j^\xi - M(1-x_{i,j}) && \forall (i,j)\in A, && \text{(1i)}\\
& a_i^\xi + s_i^\xi + \tau_{i,n+1}^\xi - L \leqslant g_i^\xi + \theta(1-x_{i,n+1}) && \forall i\in[\![1,n]\!], && \text{(1j)}\\
& h_i^\xi \geqslant (t_i - W) - a_i^\xi && \forall i\in[\![1,n]\!], && \text{(1k)}\\
& w_i^\xi \geqslant a_i^\xi - (t_i + W) && \forall i\in[\![1,n]\!], && \text{(1l)}\\
& t_i \leqslant L, \;\; g_i^\xi \leqslant \theta && \forall i\in[\![1,n]\!], && \text{(1m)}\\
& a_i^\xi, h_i^\xi, w_i^\xi, g_i^\xi \geqslant 0 && \forall i\in[\![1,n]\!]. && \text{(1n)}
\end{aligned}
$$

Scenario-based objective function (1g) minimises the idle, wait and overtime costs. The functionality of (1h) is two-fold. First they join (1i) to link together the arrival time to the first customer, its service time, and the traversal time to the next customer given that the two customer visits are consecutive. Second it forbids the formation of subtours, which are circles

formed only by a group of customers without the depot. (1j) determine the incurred overtime when returning to the depot from the last customer. Constraints (1k) and (1l) specify idle and waiting times, respectively. Constraints (1m) give the upper bounds, and (1n) provide lower bounds for the relevant variables.

3 Exact Solution Method

3.1 Deterministic Exact Solution Method

The deterministic model can be considered as a single-scenario stochastic model, where appointment time t_i is the same as the arrival time a_i with zero service team idle time $h_i = t_i - a_i = 0$ at customer $i \in [\![1, n]\!]$. Besides, the model has a pre-specified set of customer nodes with known coordinates, since we assume all cancelled customers are already removed. We first attempted to solve the deterministic *H-SARA* model to optimality. The first deterministic model (first two rows) in Table 2 shows the average CPU times and the average gap solved using CPLEX 20.1.0 for 10 iterations with time limit 1800 seconds. The gap indicates the solution's quality and is defined as the difference in percentage between the upper and the lower bounds.

Type		**Number of customers** n				
		10	15	20	25	30
Deterministic	CPU-time	0.54	1.14	4.97	18.81	1800*
	Gap	0%	0%	0%	0%	0.05%
Deterministic[1]	CPU-time	0.65	1.34	4.41	16.65	1438.16
(fixed m)	Gap	0%	0%	0%	0%	0%
Deterministic[2]	CPU-time	0.63	1.31	2.07	3.62	6.33
(fixed m+ gap)	Gap	0.1%	1.7%	1.6%	0.9%	1.3%

Table 2: Results for deterministic version of the model

Although solving a smaller-scaled deterministic problem is computationally manageable, the solver fails to find feasible solutions for large or even moderate-sized instances within 30 minutes on average, as shown in Table 2. As a result, we have proposed two acceleration approaches to reduce the computing time for the deterministic *H-SARA* model. The approaches are implemented inside our solution framework and are described below.

First, we apply a root node solution method to address a trade-off between fixed vehicle costs and variable routing costs, aiming for an "economical fleet size" [Gol+84]. We pre-define the number of service teams m in constraint (1b) and change its sense to strict equality so that the solver is no longer required to optimise the fleet size but treats it as an input parameter. To find this optimal number of service teams, we limit our fleet size options by computing the upper bound ℓ_u and lower bound ℓ_l of a feasible number of service teams to hire, using the linear models proposed by [Gut+19]. The detailed models for computing ℓ_u and ℓ_l can be found in §3.1.1. For each fixed fleet size m in $\{\ell_l, \ell_l + 1, \ldots, \ell_u\}$, we used CPLEX to obtain the feasible (integer) solution at the root node. After all associated root node values are computed, we instruct CPLEX to identify the smallest root node value and return its associated m, which will be used as the final fleet size to solve the model to optimality. Using this method, we observe a considerable improvement in computation speed without loss of solution quality, as shown in the third and fourth row of Table 2.

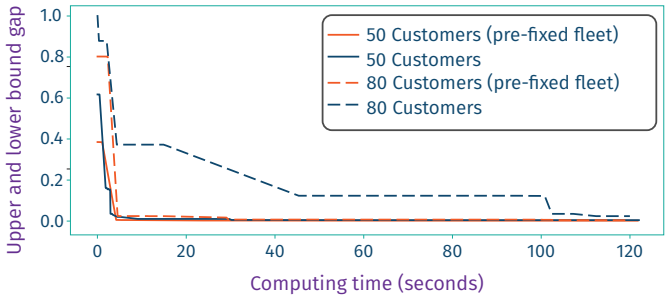

Figure 1: Deterministic model gap versus computing time

Secondly, we observe from experimental testings that CPLEX's default heuristic solution method can reach an integer solution at the root node with reasonably good quality and within a concise computing time (less than 1 minute). Nevertheless, reaching a global optimum is difficult due to the time-consuming nature of the branch-and-bound process encoded in the solver. This trend is shown in Figure 1 and can be observed visually during the solution process that the solver spends an awfully long time improving the visiting sequence of customers. Therefore, to further accelerate the solution method, we terminate the solving process if the gap is less than 2%. The solution time and relative MIP gap reported by CPLEX for different customer sizes are presented in the last two rows of Table 2.

3.1.1 Bounding the number of service teams

This section presents the upper and lower bounds of a feasible number of service teams to hire. For notation simplicity, the travel and service times involved in the models are the expected values. Following the steps given in [Gut+19], we can find an upper bound on the number of teams required to visit all clients by solving

$$\min_{\ell_u} \quad \ell_u \tag{2a}$$

$$\text{s.t.} \quad \sum_{i\in[\![1,n]\!]} s_i + \sum_{i\in V} \Big(\max\{\tau_{i,j} : (i,j)\in A\} + \max\{\tau_{j,i} : (j,i)\in A\}\Big) \leqslant \ell_u(L+\theta), \tag{2b}$$

$$1 \leqslant \ell_u \leqslant \hat{m}, \qquad \text{and} \qquad \ell_u \in \mathbb{Z}, \tag{2c}$$

where ℓ_u is a decision variable representing the maximum number of needed teams to satisfy, in a worst-case scenario, all the transportation and services requirements. Here, $\hat{m}$ is an upper limit on the number of teams, which can be as large as the number of customers n, and an optimal solution of (2) determines a choice over m. Observe that if we divide constraint (2b) by ℓ, the resulting expression distributes the routing task in two parts: There is a term averaging service time, and another term averaging the time required, taking time-consuming paths, to travel between customers. Notice that the optimal solution can be obtained using exhaustive enumeration in $\mathcal{O}(\hat{m})$ time.

Likewise, service times can provide a lower bound on the amount of time that all service teams spend on the road. To do so, we define ℓ_l as the minimum number of teams required to distribute the aggregated service time and minimum transportation time. Thus, we solve

$$\max_{\ell_l} \quad F(\ell_l) = \sum_{i\in[\![1,n]\!]} \frac{s_i}{\ell_l} + \sum_{i\in V} \left[\frac{\min\{\tau_{i,j} : (i,j)\in A, i\neq j\}}{\ell_l} + \frac{\min\{\tau_{j,i} : (j,i)\in A, i\neq j\}}{\ell_l} \right] \tag{3a}$$

$$\text{s.t.} \quad F(\ell_l) \leqslant L + \theta, \qquad 1 \leqslant \ell_l \leqslant m, \quad \text{and} \quad \ell_l \in \mathbb{Z}. \tag{3b}$$

Notice that if this problem is infeasible, then there are not enough teams to solve H-SARA. As a result, we have an infeasibility certificate. Again, this problem can be solved in $\mathcal{O}(\hat{m})$ time.

3.2 Stochastic Exact Solution Method

The multi-scenario stochastic model is considerably more challenging to tackle than its deterministic counterpart. We realise the natural partitioning of our stochastic model, where the first stage is a mixed-integer linear programming problem and the second-stage recourse model is linear. Furthermore, the second-stage problem is scenario-dependent, and therefore its structure suggests the application of *Benders' Decomposition* [Ben05]. We use CPLEX built-in Benders callback to solve a full model. The first and second row in Table 3 list the numerical results of solving the complete stochastic model as a whole, incorporating the fleet size pre-solving procedure described in the deterministic model, and limiting the gap to 2%. The third and fourth row are with Benders' callback. The empirical results show that *Benders Decomposition* is not suitable for our models as it consumes much longer computing time to provide worse results.

Type		**Number of customers** n				
		5	10	15	20	30
Stochastic	CPU-time	0.19	2.05	1421.18	25.38	183.31
	Gap	1.99%	1.97%	2.12%	1.99%	1.91%
Stochastic	CPU-time	0.52	11.28	1800*	1800*	1800*
(Benders)	Gap	1.99%	2.00%	2.93%	2.12%	3.66%

Table 3: Results for stochastic model with 10 scenarios

We have observed from Table 2 and 3 that solving a large-scale H-SARA problem jointly to optimality is still not practical due to the time-consuming nature of exact methods. On top of that, the problem involves a range of uncertainties in real-life traversal times, service duration, and customer presence rates, all of which require a flexible solution method that focuses more on adapting to fast-changing information and a large number of scenarios, meanwhile achieving in-time solution with good quality. These results drove us to explore and develop a simple and flexible heuristic as an alternative.

4 Two-stage Heuristic

4.1 Solution Framework

Realising the difficulty of tackling H-SARA as a whole even with accelerating methods, we have decomposed the problem into its fleet-sizing, districting, routing, and scheduling components and developed a problem-tailored two-stage heuristic, which dynamically tackles one or multiple decisions without leaving the others out of sight. The evolution of information plays a crucial role throughout the problem, making some of the decisions dependent on the others or the realisation of data. For instance, we would not know the number of customer districts before we figure out the number of service teams to hire, or likewise, we would not be able to compute

the exact customers' appointment times until the explicit routes and customer sequences are designed.

Our two-stage heuristic resembles a typical home service rundown: previous-day initial plannings (§4.2 and §4.3), service day tour refinements (§4.4), and post-service performance evaluation (§4.5). The heuristic takes into account the evolution of information by integrating its decisions based on up-to-date data realisations at each stage. Figure 2 shows an example for our two-stage heuristic timeline, and Figure 3 displays an example.

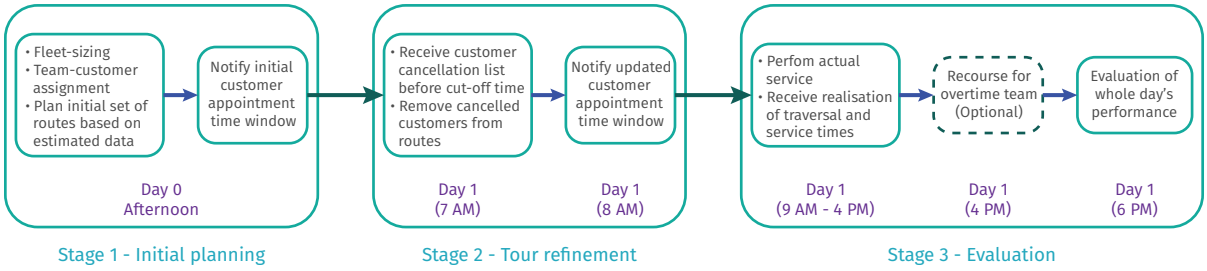

Figure 2: Heuristic rundown with chronological timeline

During the initial planning stage, the decision-makers need to make pre-arrangements with limited information to guarantee a smooth rundown on the actual service day. This stage consists of finding an initial set of routes and improving them using neighbourhood search. The tour refinement stage resembles the actual service day, with the visiting sequences re-optimised based on last-minute cancellation outcomes. For the post-service evaluation stage, complete information about travel and service durations are revealed after the actual service, allowing decision-makers to evaluate the service teams' performance. One crucial requirement for the first-stage decisions is robustness, which allows the second-stage decisions to refine the previous ones without much modification.

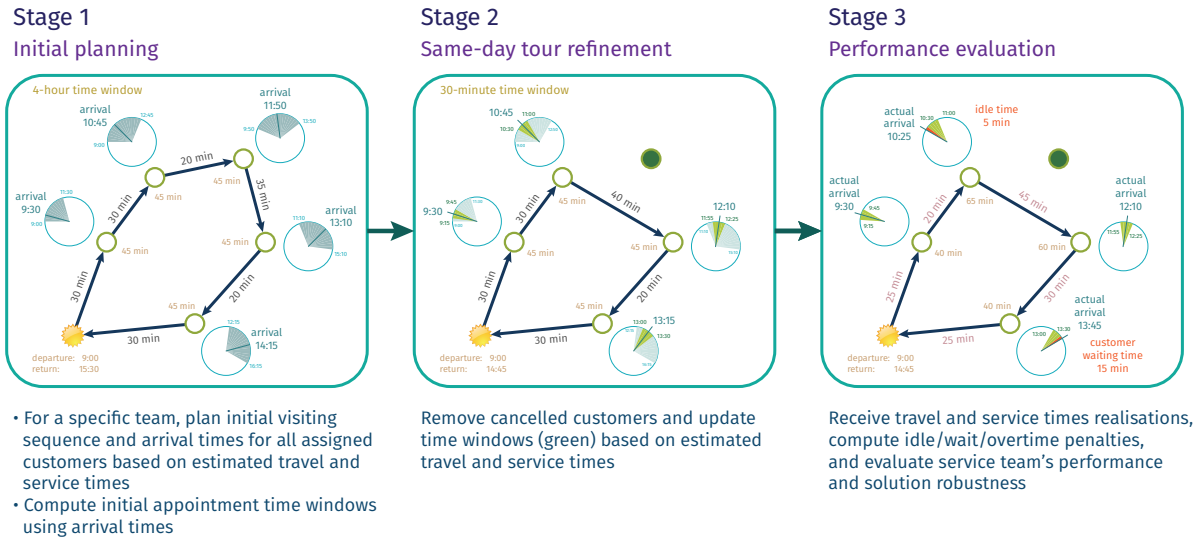

Figure 3: Heuristic framework: initial planning, tour refinement, and post-service performance evaluation stages

Before we formally introduce the two-stage heuristic, we first provide an estimation on the

activity measure, which is the expected amount of time required to include a specific customer in a tour. This helps us to determine the size of a tour serveable by an individual team. In our application, the customer cancellation rate is known probabilistically, which means that the actual sequencing of customers or the computation of route lengths is pointless without knowing the actual cancellation list. Yet, we can estimate the travel and service times without explicit routing as in [BJ09]. The estimated total time required for a group of customers can be divided into (i) *stem time*: estimated travel time from the depot to the nearest customer inside the group; (ii) *intermediate transit*: estimated travel time between customers of the same group; (iii) *service time*: estimated stopping time at each customer. Parts (i) and (iii) are self-explanatory and can be estimated by the relevant probabilistic distributions for travel and service durations. A detailed formulation for (ii) is explained below.

4.1.1 Estimate service and travel times with probabilistic customer presence

For (ii), we can estimate the expected travel time from customer i to any same-group customer j with probabilistic customer presence rate, using the formula given in [BJ09]:

$$\begin{aligned} t_i^{(*)} &= \sum_{j=1}^{n_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{v_{i,j}} \\ &= \sum_{j=1}^{n_i} \frac{(1-p_j)(n_i - R_{i,j}+1)}{\sum_{l=1}^{n_i}(1-p_l)(n_i - R_{i,l}+1)} \cdot \frac{d_{i,j}}{v_{i,j}} \end{aligned} \tag{4}$$

where p_j is customer j's probabilistic cancellation rate introduced earlier, n_i is the number of closest customers to customer i, $R_{i,j}$ is the rank of the j^{th} closest customer to i, with $j \in [\![1, n_i]\!]$, and $p_{i,j}^{(*)}$ can be interpreted as the likelihood of customer j following i on a route. $d_{i,j}$ is the Euclidean metric and $v_{i,j}$ is the travel velocity from node i to j.

As a result, the activity measure ω_i for customer i can be estimated by the estimated service time s_i plus the estimated travel time from i to the district centre j using (4). Here we use the average travel velocity $\hat{v}$. We estimate the number of nearest customers to be the average number of customers inside a district $n_i = \lceil \frac{N}{m} \rceil$. This way we have the estimated activity measure for a specific customer i:

$$\omega_i = s_i + t_i^{(*)} = s_i + \sum_{i=1}^{n_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{\hat{v_{i,j}}} \tag{5}$$

4.2 Initial Set of Routes

At the beginning of the initial planning stage, we apply a cluster-first-route-second construction heuristic to come up with an initial set of routes. A feasible fleet size m can be pre-determined using the root-node solution method we described in Section 3.1. We adapt the districting formulation proposed by Hess et al. [Hes+65] and solve the MIP model, which is explained below, to optimality to receive our initial customer-team assignment decisions.

Districting MIP Formulation Let $[\![1, n]\!]$ be the set of customers and $\{0\}$ be the depot as before. Let $\omega_i \in R^+$ be the activity measure associated with customer i. The number of districts to be formed is the same as the pre-defined number of vehicles m. The average activity measure per district is defined as $\mu = \frac{1}{m} \sum_{i \in [\![1,n]\!]} \omega_i$. We denote $s_{\min} \leqslant 100$ and $s_{\max} \geqslant 100$

as the minimum and maximum percentage of activity measures in a district, respectively. L is the maximum allowed working time. Denote by $d_{i,j}$ the travel (Euclidean) distance between customers i and j. Finally, the decision variable $y_{i,j}$ is equal to one if customer i is assigned to the district centred at customer j, and it is zero otherwise. Here $y_{j,j}$ takes the value of one if customer j is selected to be the district centre.

The districting MIP model can be defined as below:

$$\min_{y} \sum_{j\in[\![1,n]\!]} \sum_{i\in[\![1,n]\!]} \omega_i d_{i,j}^2 y_{i,j} \tag{6a}$$

$$\text{s.t.} \sum_{j\in[\![1,n]\!]} y_{i,j} = 1 \qquad \forall i \in [\![1,n]\!] \tag{6b}$$

$$\sum_{j\in[\![1,n]\!]} y_{j,j} = m \tag{6c}$$

$$y_{i,j} \leqslant y_{j,j} \qquad \forall j \in [\![1,n]\!] \tag{6d}$$

$$\sum_{i\in[\![1,n]\!]} \omega_i y_{i,j} \geqslant \frac{s_{\min}}{100}\mu \cdot y_{j,j} \qquad \forall j \in [\![1,n]\!] \tag{6e}$$

$$\sum_{i\in[\![1,n]\!]} \omega_i y_{i,j} + 2d_{0j} \leqslant L \qquad \forall j \in [\![1,n]\!] \tag{6f}$$

$$y_{i,j} \in \{0,1\} \qquad \forall i,j \in [\![1,n]\!] \tag{6g}$$

Constraints (6b) require every customer to be assigned to a district. Constraint (6c) requires exactly m districts to be formed. Constraints (6d) state that each formed district must have a center. Constraints (6e) define the minimal workload of any district. Constraint (6f) stresses that the workload within each district, i.e. the activity measure within each district together with the pendulum tour to and from the depot, has to be no more than the total time allowance.

Mathematically, we first aggregate customers into m compact and balanced districts that are each manageable by an individual service team. After clustering the customers, we form a single cycle inside each district containing all its customers and the depot. This is equivalent to solving the TSP for m times. We adapt the Dantzig-Fulkerson-Johnson TSP formulation to receive our initial routing decisions. A comprehensive review on the TSP heuristics methodologies and implementations can be found in [Reg+11]. However, considering the size of our problem, an exact solution can be obtained using existing solvers.

4.3 Neighbourhood Search

To improve upon the initial set of routes, we employ the *adaptive large neighbourhood search* (ALNS) meta-heuristic. ALNS was first introduced by Ropke and Pisinger [RP06] as an extension of the *large neighbourhood search* (LNS) proposed by Shaw [Sha99] with the general principle of "destroy and repair", which is to search for a better solution by destructing a part of the solution and reconstructing the damaged part in a different way. Our ALNS pseudocode is presented in Algorithm 1.

The upper set of graphs in Figure 5 shows an example for the first-stage (initial planning) heuristic outputs. In Stage 1.3 in Figure 5, we further balance the workload among all operators by including a workload imbalance penalty in the ALNS objective function to penalise the extra units of workload above or below a certain threshold for any service team. The last step of the first-stage heuristic is to notify all customers of their initial appointment slots. Based on

Algorithm 1 Basic steps of ALNS

$s \leftarrow$ `InitialSolution`, `InitialScore`(w^*) and $s^{\text{best}} = s$
for `stopping criteria not met` **do**
  $N^- \leftarrow$ `Choose`(`AllDestroyOperators`, w_d^*)
  $N^+ \leftarrow$ `Choose`(`AllRepairOperators`, w_r^*)
  $s' \leftarrow$ `DestroyRepairApply`(s, N^-, N^+)
  if $s' <$ `QualityThreshold` **then**
    $s' \leftarrow$ `LocalSearch`(s')
  `obj`(s') = `sum cost (team, travel, overtime) and workload balance penalties`
  if s' `satisfies acceptance criterion` **then**
    $s \leftarrow s'$
    if $s' < s^{\text{best}}$ **then**
      $s^{\text{best}} \leftarrow s'$
  update `RouletteWheel operators performance scores`

the set of routes improved by ALNS, we compute each individual's appointment time from the associated team departure time. To cope with potential last-minute customer cancellations, we expand each individual appointment time into an appointment time window with fixed length and communicate this individual-tailored appointment time window to every registered customer. For example, assuming $T_1 = 4$ hours and a customer's estimated appointment time is at 11:30 am, the first-stage appointment time window for them will be [9:30, 13:30].

4.3.1 Destroy and Repair Operators

The algorithm removes a pre-defined number of nodes from the solution together with their linking arcs before adding them back iteratively, with the hope that the newly formed solution yields a smaller objective value. We introduce the whole list of destroy operators below:

1. *Random Removal:* a group of q randomly selected customers are removed from their existing routes and placed inside the customer pool.
2. *Worse Removal:* originally proposed in [RP06] where the q customers with the highest removal gain, which is the difference in cost when this customer is inside an allocated tour, and when the customer is not.
3. *Related Removal:* a single customer is randomly selected and removed together with the $(q-1)$ nearest customers from their tours and added to the customer pool.
4. *Tour Removal:* Randomly remove a single tour departing from the facility. Move all the allocated customers from this single tour to the customer pool.
5. *Longest Tour Break into Half:* Break the longest tour found into two smaller tours. Link the start and end of the smaller tours to the depot.
6. *Overcapacitated Tour Break into Half:* Break all the infeasible tours (time capacity violated) in the middle and form two smaller tours. Link the start and end of the smaller tours to the depot.

The first three destroy operators are at the customer node level, and the latter three are at the routing level. We set default $q = 5$. Customers inside the customer pool will be re-inserted by an repair operator selected from below [HCC12]:

1. *Greedy Insertion:* Randomly select a customer from the customer pool, insert it into the position that increases the total expected costs by the least. The insertion can be between two consecutive customers or between the depot and a linking customer.

2. *Greedy Insertion Perturbation:* The same mechanism as *Greedy Insertion*. However, the insertion cost of the selected customer at each specific position is influenced by a perturbation factor d between [0.8, 1.2].
3. *Greedy Insertion Forbidden:* The same mechanism as *Greedy Insertion*, only that a customer node cannot be re-inserted to the same position removed from.

Since destroy and repair operators (with local search) allow us to modify the number of existing tours, it is possible to re-optimise the fleet size during the ALNS search. As a result, ALNS allows our first-stage heuristic to be less affected by a poor selection of service team number m at the beginning.

4.3.2 Local Search

Local search methods (*move*, *swap*, and *2-opt*) are applied after each destroy-repair iteration to further improve the repaired solutions. However, since local search is usually computationally expensive, we only wish to apply it to promising candidates whose objective values after the repair stage are within a limit of the best-found incumbent (default 30%). A graphical description of the *move*, *swap*, and *two-opt* methods is given in Figure 4.

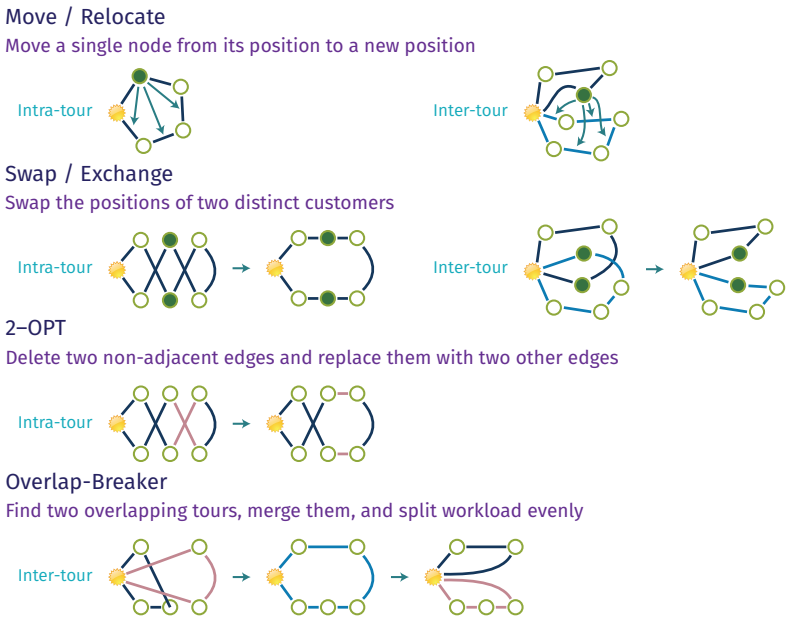

Figure 4: Local Search Operators

4.3.3 Roulette Wheel Selector with Adaptive Weight

We apply the *roulette wheel* (a probabilistic mechanism) to select the destroy and repair operator pair at each iteration based on each operator's weight, which dynamically changes throughout the algorithm. The selector goes like this: each time a selected operator i brings improvement to the incumbent, a score σ (dependent on the solution quality) is added to this operator's total weight $w_i^{(*)}$. We define the probability of this operator being selected for the next iteration as $w_i^{(*)} / \sum_{k=1}^{K} w_k^{(*)}$, with K being the number of all destroy or repair operators. *Roulette wheel with adaptive weight adjustment* was first proposed in [RP06] to avoid a specific destroy or repair

operator overwhelmingly dominating the whole search process. Here, the associated weight of an operator is computed using two terms: the performance from the past N iterations and the whole performance overall. The operator's weight can be mathematically computed as below:

$$w_{i,j+1}^{(*)} = \begin{cases} w_{i,j}^{(*)}(1-r) + \frac{\pi_i}{Q_i} r & \text{if } \pi_i > 0, \\ w_{i,j}^{(*)} & \text{if } \pi_i = 0, \end{cases} \tag{7}$$

where $w_{i,j}^{(*)}$ is the weight of operator i used in segment j, π_i is the score of operator i obtained in the previous segment, Q_i is the number of times an operator i is employed during the previous segment, and r is the reaction factor that controls how strongly each operator performance is determined by the last segment and the overall process. Here we choose $r = 1/2$.

4.3.4 Acceptance and Stopping Criteria

ALNS has an embedded *simulated annealing* (SA) as the acceptance criterion that allows the algorithm to accept a newly-found solution s' not necessarily yielding overall lower costs. SA contributes to ALNS's strong capability and robustness in exploring the solution neighbourhood with both diversification and intensification, allowing the search to escape from a local minimum and visit unexplored areas of the search space. Mathematically, we accept the new solution s' with probability $e^{f(s')-f(s)/T}$ where s is the current solution and T the initial temperature.

For the stopping criteria, we force the search to be terminated after either a certain amount of time or a prescribed number of non-improving iterations is reached.

4.3.5 Further Improvements

To further improve on the real-life practicality of our routes and schedules derived after the districting-first-routing-second construction heuristic of §4.2 and ALNS improvement heuristic of §4.3, we have considered the following improvements for our first-stage solution: workload balance between teams, multiple tours overlapping minimisation, and single tour self-intersection elimination.

Each service team's assigned workload is bounded by (6e) and (6f), which means a team could still be assigned a much higher or lower workload compared to the rest of the teams. To further balance the workload among all, we include a soft workload balance penalty $P \cdot \max\{|\frac{\omega(D_k)-\mu}{\mu}| - \alpha, 0\}$ in the ALNS objective function to penalise the extra units of workload above or below a certain threshold for any service team. We have chosen $\mu = 0.3$ based on experimental results.

Occasional multiple tours overlapping is unavoidable, especially with a tight number of available service teams. Service durations have a larger scale than the inter-customer travel times, leading to customer assignments prioritising a good fit of customer service times into the remaining workload over the geographical adjacency. The randomness of customer geographical location can result in an unevenly high concentration of customers, challenging for the algorithm to form disjoint, compact, and contiguous driver zones within a reasonable computing time. However, the application of *overlap-breaker* or *2-opt* (Figure 4) can remove the majority of overlaps and eliminate tour self-intersections.

Moreover, we keep a "safety time" $L_\delta = 30$min from the assigned standard service hours L in our computation. Consequently, each service team will have a slightly lower utilisation rate, but this will result in a higher risk-averse level and a lower chance of working overtime due to factors such as traffic congestion and service delay.

4.4 Tour Refinements Stage

At the beginning of the second stage (on the service day), the list of cancelled customers is revealed. The second stage model re-optimises the previously made decisions to fit the up-to-date customer information. The lower set of graphs in Figure 5 show an example for the second-stage tour refinement: we remove cancelled customers from the previously scheduled tours, compute the new estimated arrival times for all non-cancelled customers, apply routing improvements to guarantee service team workload balance, and notify all non-cancelled customers of an updated narrower appointment time window. We re-apply the ALNS improvement heuristic in Stage 2.3, where we not only minimise the total travelling costs, overtime costs, and team workload imbalance but maximise the chance of scheduling the updated appointment times to nest within the first-stage appointment time windows. In this way, we avoid abrupt appointment time modifications, which is essential to service quality and customer satisfaction.

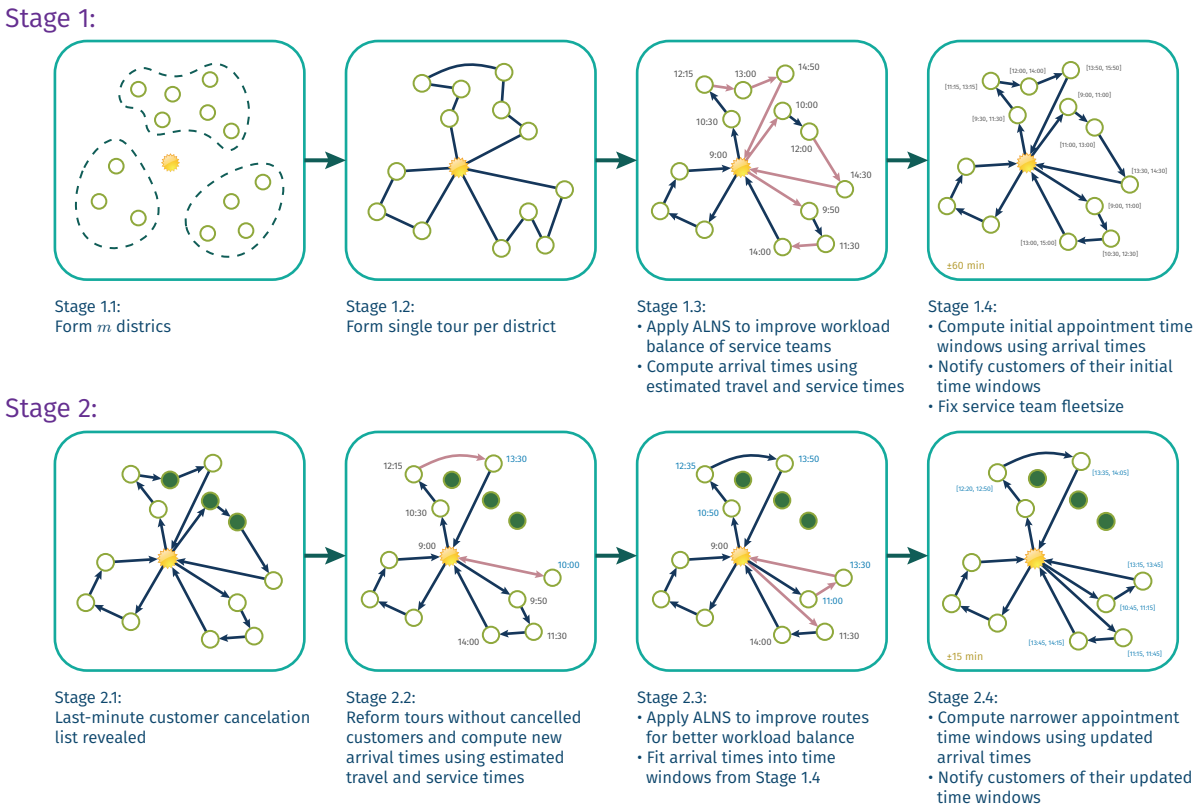

Figure 5: Initial planning and route refinement stages of the heuristic framework - an example

The service teams' arrival times to customers and depot are random variables since they depend on travel and service times which are by definition random variables. This lead to our decision of quoting an appointment time window, rather than a specific time point, to every non-cancelled customer during the first and second stages. Having the second-stage time window nested within the first-stage time window is essential. Specifically, we assume a first-stage time window $[T_1^{\text{start}}, T_1^{\text{end}}]$ and a second-stage estimated arrival time a_i at a non-cancelled customer i. The ALNS objective term $P' \times \max\{T_1^{\text{start}} - a_i, a_i - T_1^{\text{end}}, 0\}$ penalises any arrival time not nested within the first-stage time window. Similar to the first-stage appointment scheduling, we create a narrowed second-stage time window with length $T_2 = 30\text{min}$. The time windows

are not necessarily centred at their arrival times. This is determined by a linear adjustment $[a_i - T_i^{start}] * c_{idle} = [T_i^{end} - a_i] * c_{wait}$ that forces the center forward in time to cope with more expensive waiting costs, or backward with more expensive idle costs.

4.5 Post-Service Performance Evaluation

The quality of our second-stage refined routes will be evaluated in the post-service evaluation stage. The issue of data over-fitting might occur for our two-stage heuristic, since we only rely on in-sample objective values computed using a discretised set of scenarios n_e clustered from random samples. Therefore, we also evaluate the out-of-sample performance of our solutions using a new and much larger set of benchmark scenarios generated after the model has been solved. This gives a fairer indication of how good our service levels are with an unobserved set of data.

5 Experiments and Insights

In this section, we present our computational results. For a unified measurement, we use CPLEX 20.1.0 as the optimisation solver for both the heuristic framework and exact methods. The whole two-stage heuristic solution computation is performed on a machine with Intel i5-10400F CPU and 16GB RAM installed. To compare the stochastic *H-SARA-2* model with our two-stage heuristic on an equal footing, we add two extra type of constraints $h_i^{\xi} \geqslant T_i^{start} - a_i^{\xi}$ and $w_i^{\xi} \geqslant a_i^{\xi} - T_i^{end}$ based on appointment time window $[T_i^{start}, T_i^{end}]$ assigned in Stage 1.1.

Since the cancellation list is random, and so are the travel and service durations, we come up with a sampling-based objective function computed from a number of n_e randomly generated scenarios to guide the second-stage solution process. We introduce a scenario generating procedure to ensure a more diverse set of scenarios is included. First we apply the Monte-Carlo simulation that randomly generates n_s samples, each with an identical pair of travel and service times realisations. We then cluster a fixed number of n_e scenarios from these samples using a k-mean clustering algorithm given that $n_s \gg n_e$. The probability q^{ξ} of each scenario ξ is estimated using the number of samples clustered together divided by the total number of generated samples. In this way, we are able to capture extreme values using a moderate number of scenarios.

The experiment results are given in Table 4. Score is the expected objective function value averaged from 10 experiments on 100 test instances generated out of 10,000 samples by k-means. Time is measured in seconds, and t^* refers to the upper limit of computing time which is 1800 s. All the results (Score and Time) are averaged from 10 experiments. Scenarios used in methods are generated randomly and independently from test instances. SVRP is solved with the proposed time-saving root node solution strategy (pre-selection of fleetsize m) given in § 3. 2-Stage Heur is solved using our two-stage heuristic method with the tour-overlap-breaker local search operator in the ALNS improvement process. 2-Stage ALNS is solved using our proposed two-stage heuristic method without the tour-overlap-breaker local search operator in the ALNS improvement process. This model applies the classical ALNS, which is included here for comparison with our improved ALNS in both speed and outcome. "1-stage" Heur is solved as a comparison to our two-stage heuristic, assuming the full customer cancellation list being available before the initial planning stage. Thus the second-stage re-routing and re-scheduling are excluded from the solution process. We solve this by not removing any customers at the second stage. This comparison tells how much last-minute customer cancellation costs to the business apart from other uncertainties.

# Scenarios	1								10				20				50				100			
Method	SVRP		2-stage Heur		2-stage ALNS		"1-stage" Heur		SVRP		2-stage Heur		SVRP		2-stage Heur		SVRP		2-stage Heur		SVRP		2-stage Heur	
Metrics	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time
$n = 10$	209.23	1.638	169.42	1.28	168.79	1.27	169.07	0.81	211.04	1.93	172.54	2.49	209.10	2.29	164.03	3.84	208.19	3.47	161.28	7.84	205.94	13.20	162.74	14.08
$n = 20$	236.65	34.55	235.51	15.75	238.43	15.83	234.56	8.16	237.35	192.46	237.71	19.96	-	t*	240.46	24.64	-	t*	237.15	36.56	-	t*	235.76	57.73
$n = 30$	315.68	364.54	318.77	19.87	318.96	19.87	315.64	15.32	-	t*	318.83	27.68	-	t*	319.55	35.77	-	t*	318.08	50.12	-	t*	318.95	91.42
$n = 40$	410.30	1115.24	418.28	35.21	418.72	35.48	409.82	23.52	-	t*	417.54	47.04	-	t*	424.14	58.98	-	t*	417.62	95.54	-	t*	417.62	157.54
$n = 50$	-	t*	521.91	72.87	522.50	73.98	512.59	50.90	-	t*	521.37	88.80	-	t*	521.64	105.65	-	t*	521.82	154.27	-	t*	521.22	239.22
$n = 100$	-	t*	1012.28	198.84	1016.24	199.51	1020.43	173.96	-	t*	1018.09	256.23	-	t*	1021.63	310.54	-	t*	1004.83	459.90	-	t*	998.48	790.11
$n = 150$	-	t*	1459.51	609.81	1465.49	612.87	1504.49	586.06	-	t*	1460.89	716.40	-	t*	1460.82	827.44	-	t*	1461.24	1148.10	-	t*	1460.78	1645.79

Table 4: Computational results from different solution methods

From this table, we have observed the following points. To begin with, our two-stage heuristic provides competitive solutions comparing to CPLEX solutions on the same set of simulated benchmark instances. By comparing same-scenario columns between exact method and heuristic, we observe that within the given time limit, the two-stage heuristic is able to find solutions within 4% of the solutions computed by CPLEX. Even though all exact and heuristic methods columns are non-optimal (since global optimum is extremely difficult to compute, as shown in Fig 1), we want to showcase the fact that our two-stage heuristic is able to provide same-quality solutions and within less amount of time compared to CPLEX. Besides, the two-stage heuristic is more robust in real-life applications and can provide up-to-date decisions at different service preparation stages based on different levels of available information. Our two-stage heuristic can tackle a larger customer size within a reasonable time. It takes no more than 2 minutes on our computer to compute a solution for a 40-customer instance, whereas the deterministic model requires 19 minutes on average and the stochastic model cannot even terminate within 30 minutes. If we further increase the model's size to 100 customers, none of the exact MIP approaches can terminate in hours but our two-stage heuristic can still obtain results in 5 min.

Hypothetically, if we obtain the complete customer cancellation information in the first place, we can simply merge the two heuristic stages and deal with only stochastic travel and service times. To determine the additional cost of making multi-stage decisions, we run a parallel experiment "1-stage Heur", assuming complete information for cancelled customers. It achieves lower objective costs than the two-stage heuristic "2-stage Heur", which receives no customer cancellation list but only cancellation probability during the initial planning stage. Yet, our two-stage heuristic is not worse-off in terms of average objective values and computing time from the results. For experiment sets with 100 and 150 customers, "2-stage Heur" outperforms "1-stage Heur" in the expected objective function value although with slightly longer computing time on average. We recognise two potential reasons behind this phenomenon: local search-based heuristics cannot guarantee the global optimum in general, and the solutions computed by "1-stage Heur" being over-fitted to the single scenario than the benchmark instances/scenarios from the evaluation stage.

We are able to include last-minute customer cancellation into our solution process and make initial decisions based on uncertain customer set, all at a reasonable additional cost. The additional cost is mainly due to our requirement to nest the second-stage narrower appointment time window within the first stage's, thus limiting the freedom to optimise the best routes and leading to slightly worse-off solutions. However, no perfect information exists in reality, and the differences between one-stage and two-stage solutions can be treated as the costs of "imperfect information", or equivalently, the costs for making a priori decisions and previous-day customer notifications without getting the complete picture.

6 Conclusion

In this paper, we studied the H-SARA problem, which integrates the fleet-sizing, assignment, routing, and scheduling problems. We proposed a stochastic MIP model for the H-SARA problem, whose deterministic and stochastic versions are solved with two accelerated methods for small-scale instances. We also developed a tailored two-stage heuristic solution method with an embedded ALNS improvement heuristic, to support a real-life decision-making process taking the evolution of information into account. Our proposed heuristic shows good performance in terms of computational time and solution quality. It also demonstrates good flexibility and robustness in adapting to multiple scenarios with different travel times, service times, and cus-

tomer cancellation rates. Using our decision support framework, we can provide high-quality fleetsizing, districting, routing, and scheduling decisions with low idle, waiting, and overtime costs, as well as two sets of customer appointment time windows, and balanced service team workload within geographically clear service zones.

Bibliography

[BJ09] J. F. Bard and A. I. Jarrah. "Large-scale constrained clustering for rationalizing pickup and delivery operations". In: *Transportation Research Part B* 43.5 (2009), pp. 542–561.

[Ben05] J. F. Benders. "Partitioning procedures for solving mixed-variables programming problems". In: *Computational management science* 2.1 (2005), pp. 3–19.

[BJO90] D. J. Bertsimas, P. Jaillet, and A. R. Odoni. "A Priori Optimization". In: *Operations research* 38.6 (1990), pp. 1019–1033.

[BRV16] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse. "The vehicle routing problem: State of the art classification and review". In: *Computers & Industrial Engineering* 99 (2016), pp. 300–313.

[Cap+18] P. Cappanera, M. G. Scutellà, F. Nervi, and L. Galli. "Demand uncertainty in robust Home Care optimization". In: *Omega* 80 (2018), pp. 95–110.

[Cis+17] M. Cissé, S. Yalçındağ, Y. Kergosien, E. Şahin, C. Lenté, and A. Matta. "OR problems related to Home Health Care: A review of relevant routing and scheduling problems". In: *Operations Research for Health Care* 13-14 (2017), pp. 1–22.

[DR59] G. B. Dantzig and J. H. Ramser. "The Truck Dispatching Problem". In: *Management science*. Management Science 6.1 (1959), pp. 80–91.

[Daw19] C. Dawson. *Royal Mail day before delivery time notifications launched*. https://tamebay.com/2019/04/royal-mail-day-before-delivery-time-notifications-launched.html. 2019.

[DPD20] DPD. *Guide to DPD*. https://www.dpd.co.uk/pdf/dpd_sales_guide_2020_v3.pdf. 2020.

[FH15] C. Fikar and P. Hirsch. "A matheuristic for routing real-world home service transport systems facilitating walking". In: *Journal of Cleaner Production* 105 (2015), pp. 300–310.

[FH17] C. Fikar and P. Hirsch. "Home health care routing and scheduling: A review". In: *Computers & Operations Research* 77 (2017), pp. 86–95.

[Gol+84] B. Golden, A. Assad, L. Levy, and F. Gheysens. "The fleet size and mix vehicle routing problem". In: *Computers & Operations Research* 11.1 (1984), pp. 49–66.

[GV13] E. V. Gutiérrez and C. J. Vidal. "Home health care logistics management problems: A critical review of models and methods". In: *Revista Facultad de Ingeniería Universidad de Antioquia* 68 (2013), pp. 160–175.

[Gut+19] S. Gutiérrez, A. Miniguano-Trujillo, D. Recalde, L. M. Torres, and R. Torres. "The Integrated Vehicle and Pollster Routing Problem". In: (2019). arXiv:1912.07356.

[Han+17] S. Han, L. Zhao, K. Chen, Z.-w. Luo, and D. Mishra. "Appointment scheduling and routing optimization of attended home delivery system with random customer behavior". In: *European Journal of Operational Research* 262.3 (2017), pp. 966–980.

[HCC12] V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic. "An adaptive large neighborhood search heuristic for Two-Echelon Vehicle Routing Problems arising in city logistics". In: *Computers & Operations Research* 39.12 (2012), pp. 3215–3228.

[Hes+65] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. "Nonpartisan Political Redistricting by Computer". In: *Operations research* 13.6 (1965), pp. 998–1006.

[HA14] S. G. Holm and R. O. Angelsen. "A descriptive retrospective study of time consumption in home care services: How do employees use their working time?" In: *BMC Health Services Research* 14.1 (2014), pp. 439–439.

[KS17] S. Küçükyavuz and S. Sen. "An introduction to two-stage stochastic mixed-integer programming". In: *Leading Developments from INFORMS Communities*, ed. by R. Batta et al. INFORMS TutORials in Operations Research. INFORMS, 2017. Chap. 1, pp. 1–27. DOI: 10.1287/educ.2017.0171.

[LM11] E. Lanzarone and A. Matta. "A cost assignment policy for home care patients". In: *Flexible Services and Manufacturing Journal* 24.4 (2011), pp. 465–495.

[Lap09] G. Laporte. "Fifty Years of Vehicle Routing". In: *Transportation Science* 43.4 (2009), pp. 408–416.

[Lin+14] C. Lin, K. Choy, G. Ho, S. Chung, and H. Lam. "Survey of Green Vehicle Routing Problem: Past and future trends". In: *Expert Systems with Applications* 41.4 (2014), pp. 1118–1138.

[LYJ19] R. Liu, B. Yuan, and Z. Jiang. "A branch-and-price algorithm for the home-caregiver scheduling and routing problem with stochastic travel and service times". In: *Flexible Services and Manufacturing Journal* 31.4 (2019), pp. 989–1011.

[May+15] P. Maya Duque, M. Castro, K. Sörensen, and P. Goos. "Home care service planning. The case of Landelijke Thuiszorg". In: *European Journal of Operational Research* 243.1 (2015), pp. 292–301.

[OAW18] J. Oyola, H. Arntzen, and D. L. Woodruff. "The stochastic vehicle routing problem, a literature review, part I: models". In: *EURO Journal on Transportation and Logistics* 7.3 (2018), pp. 193–221.

[Reg+11] C. Rego, D. Gamboa, F. Glover, and C. Osterman. "Traveling salesman problem heuristics: Leading methods, implementations and latest advances". In: *European Journal of Operational Research* 211.3 (2011), pp. 427–441.

[RRV20] M. I. Restrepo, L.-M. Rousseau, and J. Vallée. "Home healthcare integrated staffing and scheduling". In: *Omega* 95 (2020), pp. 102057–.

[Rod+15] C. Rodriguez, T. Garaix, X. Xie, and V. Augusto. "Staff dimensioning in homecare services with uncertain demands". In: *International Journal of Production Research* 53.24 (2015), pp. 7396–7410.

[RP06] S. Ropke and D. Pisinger. "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows". In: *Transportation science* 40.4 (2006), pp. 455–472.

[Sah04] N. V. Sahinidis. "Optimization under uncertainty: state-of-the-art and opportunities". In: *Computers & Chemical Engineering* 28.6-7 (2004), pp. 971–983.

[Sha99] P. Shaw. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In: *Principles and Practice of Constraint Programming: CP1998*, ed. by M. Maher et al. Vol. 1520. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, Berlin, Heidelberg, 1999, pp. 417–431.

[SC21] K. S. Shehadeh and M. Chiriki. "13th AIMMS-MOPTA Optimization Modeling Competition: Home Service Assignment, Routing, and Scheduling with Stochastic Service Time, Travel time, and Cancellation". In: *Modeling and Optimization: Theory and Applications (MOPTA)*. https://coral.ise.lehigh.edu/~mopta/competition, Date accessed: 7 Aug. 2021.

[Shi+18] Y. Shi, T. Boudouh, O. Grunder, and D. Wang. "Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care". In: *Expert systems with applications* 102 (2018), pp. 218–233.

[SS09] K. Sörensen and M. Sevaux. "A Practical Approach for Robust and Flexible Vehicle Routing Using Metaheuristics and Monte Carlo Sampling". In: *Journal of Mathematical Modelling and Algorithms* 8.4 (2009), p. 387.

[Tor+19] J. J. Torres, C. Li, R. M. Apap, and I. E. Grossmann. "A review on the performance of linear and mixed integer two-stage stochastic programming algorithms and software". Preprint at Optimization Online. 2019.

[YLJ15] B. Yuan, R. Liu, and Z. Jiang. "A branch-and-price algorithm for the home health care scheduling and routing problem with stochastic service times and skill requirements". In: *International Journal of Production Research* 53 (2015), pp. 7450–7464.

[ZW18] Y. Zhan and G. Wan. "Vehicle routing and appointment scheduling with team assignment for home services". In: *Computers & Operations Research* 100 (2018), pp. 1–11.

[ZWW21] Y. Zhan, Z. Wang, and G. Wan. "Home service routing and appointment scheduling with stochastic service times". In: *European Journal of Operational Research* 288.1 (2021), pp. 98–110.