




Solving the Home Service Assignment, Routing, and Appointment Scheduling (H-SARA) Problem with Uncertainties

Syu-Ning Johnn  

School of Mathematics, The University of Edinburgh, UK

Yiran Zhu  

School of Mathematics, The University of Edinburgh, UK

Andrés Miniguano-Trujillo  

The University of Edinburgh, UK

Heriot-Watt University, Edinburgh, UK

Maxwell Institute for Mathematical Sciences, Edinburgh, UK

Akshay Gupte  

School of Mathematics, The University of Edinburgh, UK

Abstract

The Home Service Assignment, Routing, and Appointment scheduling (H-SARA) problem integrates the strategic fleet-sizing, tactical assignment, operational vehicle routing and scheduling problems 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 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, to create reasonable routes with balanced workload and team-based customer service zones, and to increase customer satisfaction by introducing a two-stage appointment notification system updated at different time stages 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.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Mathematics of computing → Combinatorial algorithms; Applied computing → Transportation; Mathematics of computing → Probabilistic algorithms

Keywords and phrases Home Health Care, Mixed-Integer Linear Programming, Two-stage Stochastic, Uncertainties A Priori Optimisation, Adaptive Large Neighbourhood Search, Monte-Carlo Simulation

Digital Object Identifier 10.4230/OASICS.ATMOS.2021.4

Related Version *Previous Version:* http://www.optimization-online.org/DB_HTML/2021/07/8479.html

1 Background

Model Introduction. The home service industry constitutes businesses whose primary purpose is to provide services to people in their homes. Home services cover various sectors, including home healthcare, banking service, home beauty care, appliance repairs, home maintenance, and more. The typical requirements for the business providers are to decide



© Syu-Ning Johnn, Yiran Zhu, Andrés Miniguano-Trujillo, and Akshay Gupte; licensed under Creative Commons License CC-BY 4.0

21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021).

Editors: Matthias Müller-Hannemann and Federico Perea; Article No. 4; pp. 4:1–4:21



OpenAccess Series in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on the number of professional service teams to deliver services to geographically distributed customers, the assignment of the service teams to customers, the sequences of customer visits, and the scheduling of appointment time-slots to all customers with service demand. These specific decisions form the Home Service Assignment, Routing, and Appointment scheduling (H-SARA) problem, which is related to a set of widely studied problems in both academia and industry and was presented for the 13th AIMMS-MOPTA Optimization Modeling Competition [34]. The first is the *Vehicle Routing Problem* (VRP) which is a generalisation of the well-known *Travelling Salesman Problem* (TSP). 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, or 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 [4, 23, 24]. Whereas a VRP minimises the total routing costs using a predetermined number of vehicles or service teams (typically of homogeneous type), a *fleet-sizing problem* (FSP) minimises both the total routing costs and the economical fleet size [13], addressing a trade-off between fixed vehicle costs and variable routing costs.

Scheduling usually refers to the chronological allocation of tasks to workers such that the list of tasks (components) are accomplished within the shortest amount of time and with the 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 the 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 the customer-imposed time requirements. Some related problems are the *Appointment Scheduling Problem* (ASP) in the context of healthcare [14] and the *Home Health Care Routing and Scheduling Problem* (HHC-RSP) [6, 11].

Model Uncertainties. 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 [27]. The SVRP is usually solved by applying (two-stage) stochastic programming techniques [21, 32]. *A priori* optimisation [3] works on real-world applications in which randomness is a major concern. It applies the two-stage strategy: an initial solution is first 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 [9], Royal Mail [8]) 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 service 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 [12], and that the driving time typically accounts

for 18% to 26% of the total working time [20], 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 [10, 26, 29, 30, 35].

There are existing works in the literature that deal with uncertainties in travel times, service times, or customer presence in the context of *Home Health Care* (HHC). Readers are referred to [15] for reviews of relevant models and methods in HHC. An excessive studies in VRP with stochastic travel times can be found in [27]. Particularly, [22, 37, 38, 39] consider randomness in service times. [25, 35] consider travel and service times uncertainties. [5] considers customers who request service cancellation, and [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.

Our Contributions. The main contribution of this work is the treatment of an H-SARA problem integrating the four decisions levels: strategic fleet-sizing, tactical assignment, operational routing, and operational scheduling. Travel times, service duration, and cancellation rates are considered jointly as uncertain quantities, which to our best knowledge, has not been investigated in the literature before. We developed a two-stage heuristic approach that takes the evolution of information into account, thus allowing decision-making based on imperfect information before the actual customer demands are revealed, and updating existing solutions with an increased level of information. This paper is based on the authors' submission to the AIMMS-MOPTA Modeling Competition [34] at which they were awarded the First Prize.

2 Problem Statement

Let a service area be represented by a directed graph $G := (V, A)$. Here the node set V encloses the customer set $\llbracket 1, n \rrbracket := \{1, \dots, n\}$, 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 computed using the Euclidean metric; namely $A := \{(i, j) : i \neq j, \forall i, j \in \llbracket 1, n \rrbracket\} \cup \{(0, j) : \forall j \in \llbracket 1, n \rrbracket\} \cup \{(i, n + 1) : \forall i \in \llbracket 1, n \rrbracket\}$. 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 customers and the service teams. Customers must be informed of their appointment times (or time slots) on the service day before the cut off time (8 am) or the departure of the assigned service teams from the depot, whichever is earlier. Lastly, the probability distributions associated with travel and service times are known and assumed to be independent.

3 Mixed Integer Programming Model

3.1 Uncertainties inside the model

We apply *a priori* optimisation, where a set of *a priori* vehicle routes is first planned in the presence of estimated expected travel 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 to repair a failed (infeasible) solution.

A extension beyond the consideration of stochastic travel and service times is the stochastic customer behaviour (customer presence). An option provided by Sørensen and Sevaux [36] 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. Base on this assumption, if the customer in position i is removed, the service team will travel directly from customer $i - 1$ to customer $i + 1$.

3.2 Stochastic MIP model

Parameters. We introduce the MIP formulation for the *H-SARA* and derive a set Ξ of different scenarios ξ , each associated with a different realisation of the travel and service durations with a certain probability q_ξ . 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. Let $p : \llbracket 1, n \rrbracket \rightarrow \mathbb{R}_{\geq 0}$ be a probability mass function defined over the set of customers, such that for each customer $i \in \llbracket 1, n \rrbracket$ the probability of last-minute service cancellation of i is given by p_i . The cost of hiring a homogeneous team $i \in \llbracket 1, m \rrbracket$ is taken as a constant f_m . The maximum allowed working time is given by $L \geq 0$. 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 $\theta > 0$. Any additional time beyond the maximum working time L and within $L + \theta$ results in overtime cost. Finally, let c_{wait} , c_{idle} , and c_{over} be fixed non-negative unit waiting, idling, and overtime costs, respectively.

Decision Variables. 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 $0 \leq a_i \leq L$ for the team’s arrival time at customer $i \in \llbracket 1, n \rrbracket$. Likewise, w_i and h_i are non-negative real-valued variables for the customer’s waiting time, and service team’s idling time at customer $i \in \llbracket 1, n \rrbracket$, respectively. g_i is a real-valued variable measuring the overtime of a service team, registered at their arrival at the depot when returning from customer $i \in \llbracket 1, n \rrbracket$. Finally, since an actual arrival time under the stochastic setting could be different from a customer’s initial appointment time, we have differentiated an appointment time (scheduled service start time) variable t_i for each customer $i \in \llbracket 1, n \rrbracket$. 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 an arrival after the time window leads to the customer waiting.

We have the traversal variables $x_{i,j}$ (also fleetsize, if we treat the total number of edges linking customers with the depot as twice the fleet) and the appointment time variables t_i as our first-stage decisions. In contrast, the team idling time h_i , overtime g_i , and customer waiting time w_i are our second-stage decisions dependent on the different scenarios. The first and second stage formulations for the stochastic *H-SARA* model are as follows:

(1st Stage)

$$\min_x f_m \sum_{i \in [1, n]} x_{i, n+1} + \sum_{(i, j) \in A} d_{i, j} x_{i, j} + \mathbb{E}[Q(x, \xi)] \quad (1a)$$

subject to

$$\sum_{i \in [1, n]} x_{0, i} \leq \hat{m}, \quad (1b)$$

$$\sum_{i \in [0, n]} x_{i, j} = 1 \quad \forall j \in [1, n] \quad (1c)$$

$$\sum_{i \in [0, n]} x_{i, j} = \sum_{i \in [1, n+1]} x_{j, i} \quad \forall j \in [1, n] \quad (1d)$$

$$\sum_{i \in [1, n]} x_{0, i} = \sum_{i \in [1, n]} x_{i, n+1}, \quad (1e)$$

$$x_{i, j} \in \{0, 1\} \quad \forall (i, j) \in A. \quad (1f)$$

where $\mathbb{E}[Q(x, \xi)] = \sum_{\xi \in \Xi} q^\xi \cdot Q(x, \xi)$ 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 no more than \hat{m} homogeneous service teams departing from the depot $\{0\}$. (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 domain constraints.

(2nd Stage)

$$Q(x, \xi) = \min_{w, h, g} c_{wait} \sum_{i \in [1, n]} w_i^\xi + c_{idle} \sum_{i \in [1, n]} h_i^\xi + c_{cover} \sum_{i \in [1, n]} g_i^\xi \quad (1g)$$

subject to

$$a_i^\xi + h_i^\xi + s_i^\xi + \tau_{i, j}^\xi \leq a_j^\xi + M(1 - x_{i, j}) \quad \forall (i, j) \in A, \quad (1h)$$

$$a_i^\xi + h_i^\xi + s_i^\xi + \tau_{i, j}^\xi \geq a_j^\xi - M(1 - x_{i, j}) \quad \forall (i, j) \in A, \quad (1i)$$

$$a_i^\xi + s_i^\xi + \tau_{i, n+1}^\xi - L \leq g_i^\xi + \theta(1 - x_{i, n+1}) \quad \forall i \in [1, n], \quad (1j)$$

$$h_i^\xi \geq (t_i - W) - a_i^\xi \quad \forall i \in [1, n], \quad (1k)$$

$$w_i^\xi \geq a_i^\xi - (t_i + W) \quad \forall i \in [1, n], \quad (1l)$$

$$t_i \leq L, \quad g_i^\xi \leq \theta \quad \forall i \in [1, n], \quad (1m)$$

$$a_i^\xi, h_i^\xi, w_i^\xi, g_i^\xi \geq 0 \quad \forall i \in [1, n]. \quad (1n)$$

Scenario-based objective function (1g) minimises the idling, waiting 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 the idling and waiting times, respectively. Constraints (1m) give the upper bounds, and (1n) provide lower bounds for the relevant variables.

4 Exact Solution Method

4.1 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 following models are the expected values for each arc and customer node, namely $\hat{\tau}$ and \hat{s} . Following the steps given in [16], we can find an upper bound on the number of teams required to visit all clients by solving the following linear problem:

$$\min \ell_u \tag{2a}$$

subject to

$$\sum_{i \in \llbracket 1, n \rrbracket} \hat{s}_i + \sum_{i \in V} \left(\max\{\hat{\tau}_{i,j} : (i,j) \in A\} + \max\{\hat{\tau}_{j,i} : (j,i) \in A\} \right) \leq \ell_u(L + \theta), \tag{2b}$$

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

where ℓ_u is a decision variable representing the maximum number of needed teams to satisfy, in a mean-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_u , 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 need to solve the following nonlinear program

$$\max_{\ell_l} F(\ell_l) = \sum_{i \in \llbracket 1, n \rrbracket} \frac{\hat{s}_i}{\ell_l} + \sum_{i \in V} \left[\frac{\min\{\hat{\tau}_{i,j} : (i,j) \in A \wedge i \neq j\}}{\ell_l} + \frac{\min\{\hat{\tau}_{j,i} : (j,i) \in A \wedge i \neq j\}}{\ell_l} \right] \tag{3a}$$

subject to

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

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

4.2 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 idling time $h_i = t_i - a_i = 0$ at customer $i \in \llbracket 1, n \rrbracket$. Besides, the model has a pre-specified set of customer nodes with known coordinates, since we assume all cancelled customers are already removed. The instances are generated using a scenario-based approach specified in Section 6.1. We first attempted to solve the deterministic *H-SARA* model to optimality. The model was inputted with a pre-specified number of customer nodes. The first deterministic model (first

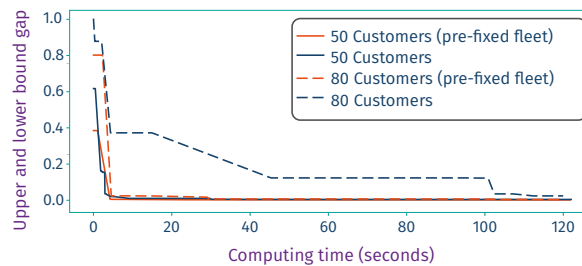
■ **Table 1** Results for deterministic version of the model with acceleration approaches.

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 (fixed m)	CPU-time	0.65	1.34	4.41	16.65	1438.16
	Gap	0%	0%	0%	0%	0%
Deterministic ² (fixed m + gap)	CPU-time	0.63	1.31	2.07	3.62	6.33
	Gap	0.1%	1.7%	1.6%	0.9%	1.3%

two rows) in Table 1 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.

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 1. 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” [13]. 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. For each fixed fleet size m in $\{\ell_l, \ell_l + 1, \dots, \ell_u\}$ computed in Section 4.1, we used CPLEX to callback the first feasible (integer) solution we receive 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 fleetsize m , which will be used as the final fleet size to optimise the routing and scheduling decisions. Using this method, we observe a considerable improvement in computation speed without a significant loss of solution quality, as shown in the third and fourth rows of Table 1.



■ **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. For an 80-customer instance, the optimal root node solution has the hiring costs outweigh routing and scheduling costs, allowing the algorithm to terminate when the idling, waiting and overtime penalties are small. Based on experimental results, we fixed a 2% gap for the total staffing, routing and scheduling costs, assessing the terminating speed (how fast to reach 2%) and solution quality (routing and scheduling decision quality). The solution time and relative MIP gap reported by CPLEX for different customer sizes are presented in the last two rows of Table 1.

4.3 Stochastic Exact Solution Method

The multi-scenario stochastic model is noticeably more challenging to tackle than its deterministic counterpart, which can be considered as a single-scenario stochastic model.

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* [2], taking the first stage as the master problem that decides which set of paths to take, and treating each scenario inside the second-stage recourse model as a subproblem. Each subproblem provides a scenario of the travel and service times for the arc traversal decisions made during the first stage.

We use CPLEX built-in Benders algorithm to solve a full model. The first and second rows in Table 2 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 rows are with Benders' algorithm. The empirical results show that *Benders Decomposition* is not suitable for our models as it consumes much longer computing time to provide worse results. Moreover, we notice that due to specific parameter scale settings, we have the fleetsize cost dominating the other costs. For a 15-customer instance, we notice that only two teams were hired, which results in seriously high idling, waiting, and overtime penalty costs. This is the reason behind the long solution process before termination, since an additional team hire brings up the total costs but is the only way to bring down the penalty costs.

■ **Table 2** Results for stochastic model with 10 scenarios.

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 (Benders)	CPU-time	0.52	11.28	1800*	1800*	1800*
	Gap	1.99%	2.00%	2.93%	2.12%	3.66%

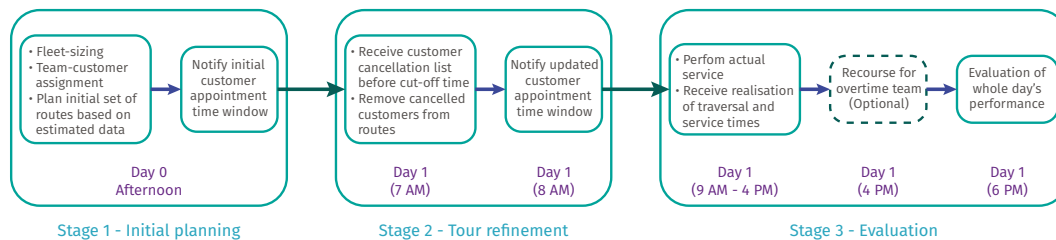
We have observed from Table 1 and 2 that even with efficient accelerating approaches, solving a large-scale H-SARA problem jointly to optimality is still not practical due to the time-consuming nature of exact methods. On top of that, the problem involves a range of uncertainties in real-life traversal 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.

5 Two-Stage Solution Strategy

5.1 Solution Framework

For our tailored two-stage heuristic, we have first decomposed the problem into different stages with an embedded chronological structure, allowing us to make dynamical decisions at each stage with an increased level of information. At each stage, we have also partitioned the decision set into its fleet-sizing, districting, routing, and scheduling components and introduced an “inter-feedback process” among different decisions, which avoids the deficiency of a hierarchical decision process that may lead to sub-optimal solutions.

Our two-stage heuristic resembles a typical home service rundown: previous-day initial plannings (Section 5.2), service day tour refinements (Section 5.3), and post-service performance evaluation (Section 5.4). The heuristic showcases the “inter-feedback process” that the previously-made decisions can be re-optimised and updated at a later stage with an increased level of information. Figure 2 shows an example for our two-stage heuristic timeline, and Figure 3 displays an example for the two-stage heuristic outputs.

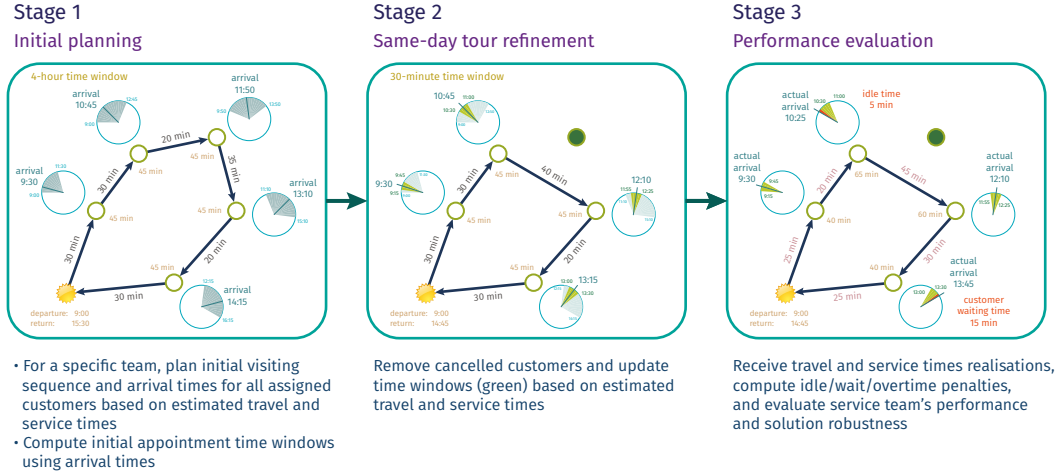


■ **Figure 2** Heuristic rundown with chronological timeline.

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. 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.

5.2 Initial Planning Stage

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 customer cluster 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 [1]. 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 travel



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

and service times distributions. For (ii), we can estimate e_i , which is the expected travel time from customer i to any same-group customer j with probabilistic customer presence rate, using the formula given in [1]:

$$e_i = \sum_{j=1}^{b_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{v_{i,j}} = \sum_{j=1}^{b_i} \frac{(1-p_j)(b_i - R_{i,j} + 1)}{\sum_{l=1}^{b_i} (1-p_l)(b_i - R_{i,l} + 1)} \cdot \frac{d_{i,j}}{v_{i,j}} \quad (4)$$

where p_j is customer j 's probabilistic cancellation rate, b_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 $j \in \llbracket 1, b_i \rrbracket$, 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 expected service time \hat{s}_i plus the estimated travel time from i to the district centre j using (4). Here we use the expected travel velocity \hat{v} . We estimate the number of nearest customers to be the average number of customers inside a district $b_i = \lceil \frac{n}{m} \rceil$. This way we have for a specific customer i :

$$\omega_i = \hat{s}_i + e_i = \hat{s}_i + \sum_{i=1}^{b_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{\hat{v}_{i,j}} \quad (5)$$

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 4.2. We adapt the districting formulation proposed by Hess et al. [19] and solve the MIP model to optimality to receive our initial customer-team assignment decisions. The specific MIP formulation can be found in Appendix A.1. 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 DFJ formulation for TSP [7] to receive our initial routing decisions. A comprehensive review on the TSP heuristics methodologies and implementations can be found in [28]. However, considering the size of our problem, an exact solution can be obtained using existing solvers.

To improve upon these routes, we employ the *adaptive large neighbourhood search* (ALNS) meta-heuristic. ALNS was first introduced by Ropke and Pisinger [31] as an extension of the *large neighbourhood search* (LNS) proposed by Shaw [33] 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. A detailed ALNS framework can be found in Appendix A.2.

■ **Algorithm 1** Basic steps of ALNS.

```

1:  $s \leftarrow \text{InitialSolution, InitialScore}(w^*)$  and  $s^{\text{best}} = s$ 
2: for stopping criteria not met do
3:    $N^- \leftarrow \text{Choose}(\text{AllDestroyOperators}, w_d^*)$ 
4:    $N^+ \leftarrow \text{Choose}(\text{AllRepairOperators}, w_r^*)$ 
5:    $s' \leftarrow \text{DestroyRepairApply}(s, N^-, N^+)$ 
6:   if  $s' < \text{QualityThreshold}$  then
7:      $s' \leftarrow \text{LocalSearch}(s')$ 
8:      $\text{obj}(s') = \text{sum cost (team, travel, overtime) and workload balance penalties}$ 
9:     if  $s'$  satisfies acceptance criterion then
10:       $s \leftarrow s'$ 
11:      if  $s' < s^{\text{best}}$  then
12:         $s^{\text{best}} \leftarrow s'$ 
13:      update RouletteWheel operators performance scores

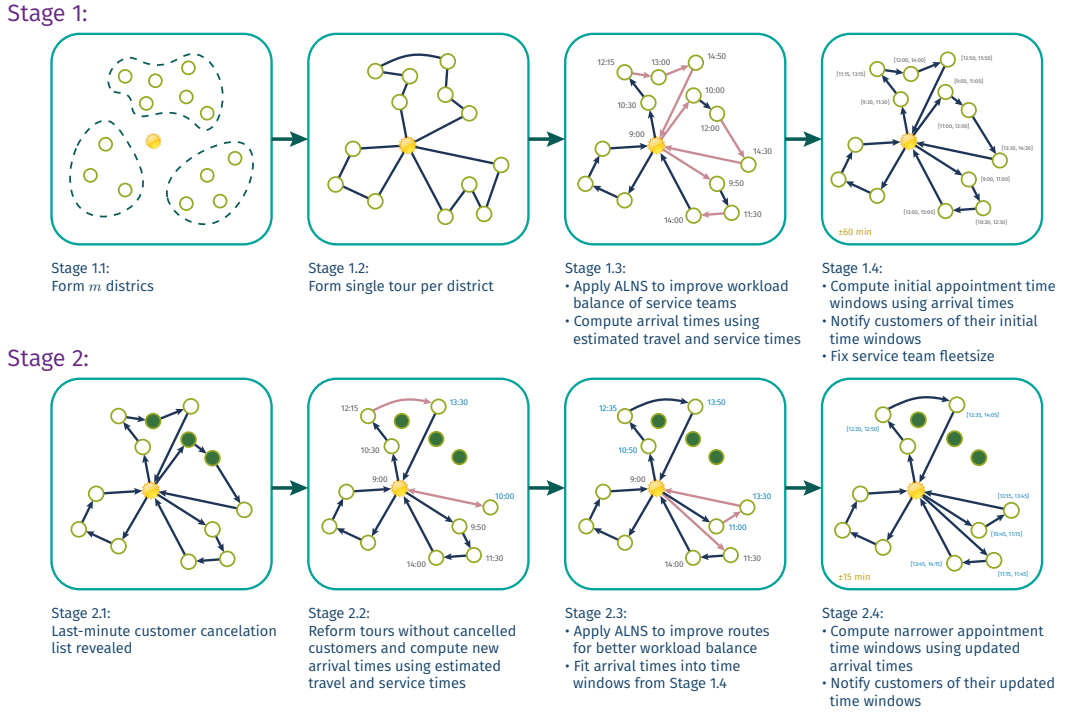
```

The upper set of graphs in Figure 4 shows an example for the first-stage (initial planning) heuristic outputs. In Stage 1.3 of the first-stage heuristic, 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 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].

5.3 Tour Refinements Stage

At the beginning of the second stage, the list of cancelled customers \mathbb{I} becomes known. So we re-optimize the initial tours to fit the updated-to-date customer information. The lower set of graphs in Figure 4 shows the decision process for our second-stage tour refinement: we remove the cancelled customers from the first-stage tours, compute the estimated arrival times for all non-cancelled customers, improve the service teams workload balance, and notify all the non-cancelled customers of a narrower appointment time window.

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. 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



■ **Figure 4** Initial planning and route refinement stages of the heuristic framework – an example.

and customer satisfaction, even though at the cost of longer service team waiting times. Specifically, we assume a first-stage time window $[T_1^{start}, T_1^{end}]$ and a second-stage estimated arrival time a_i at a non-cancelled customer i .

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 idling costs. The ALNS objective term $P' \times \max\{T_1^{start} - a_i, a_i - T_1^{end}, 0\}$ penalises any arrival time not nested within the first-stage time window. Besides, we manually adjust the second time window to be $[T_1^{start}, T_1^{start} + T_2]$ in the occurrence of any infeasible second time window begins earlier than the first. Likewise, $[T_1^{end} - T_2, T_1^{end}]$ applies to any second time window that finishes after the first time window.

5.4 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. The evaluation stage is not counted as a valid solution stage, since no decision-making process is involved.

6 Experiments and Insights

6.1 Experiment Settings

The parameter settings were given in the AIMMS-MOPTA competition guidelines [34]. Specifically, we assume n customers are uniformly located over a 50×50 km geometric grid with the depot located at the origin $(0,0)$. We set the fixed individual team hiring cost $f_m = 100$, hourly team idling time cost $c_{\text{idle}} = 2.5$, hourly overtime cost $c_{\text{over}} = 5$, hourly customer waiting cost $c_{\text{wait}} = 4$. We also define the standard daily workload $L = 8$ hours for each team, the first-stage time window length $T_1 = 2$ hours, and the second-stage time window length $T_2 = 30$ minutes. We assume the travel times between any two nodes are identically distributed with a log-normal distribution. For the customer service time, we select the gamma distribution that is not strictly symmetric in order to avoid generating a negative service time. We assume the expected service time $\hat{s} = \mu_s = 45$ min with its standard deviation set to $\mu_s/2$, the expected travel speed $\hat{v} = 1$ km/min (equivalently, expected travel time $\hat{\tau}_{i,j} = 1$ min/km). Moreover, we assume individual customers all share the cancellation probability defined at a fixed rate 5%.

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.

A Sampling-Based Objective Function. Since the customer 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, inspired by the work of [36]:

$$f^*(x) = \frac{1}{n_e} \sum_{i=1}^{n_e} f(x, S_i(\tau, s)) \quad (6)$$

where $f^*(x)$ is the expected total costs computed from a number of n_e randomly generated scenarios, x is the set of second-stage routes with the cancelled customers removed, S is the sampling function, and $S_i(\tau, s)$ represents the i^{th} scenario with stochastic travel and service times realisation. $f(x, S_i(\tau, s))$ represents the total costs of the i^{th} scenario applied to x , and finally n_e is the total size of scenarios.

Scenario Generation. 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.

6.2 Observations

The experiment results are given in Table 3, from which we have observed the following points: To begin with, our two-stage heuristic can tackle a larger customer size within a reasonable time. The two-stage heuristic 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

Table 3 Computational Results from different solution methods.

#Scenario	1						10						20						50						100					
	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		SVRP		2-stage Heur			
#Customer	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time		
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	-	-	-	-	-	-
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*	237.45	36.56	-	t*	235.76	57.73	-	-	-	-	-	-
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	-	-	-	-	-	-
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	-	-	-	-	-	-
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	-	-	-	-	-	-
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	-	-	-	-	-	-
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	-	-	-	-	-	-

* 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.

(1) SVRP is solved with the proposed time-saving root node solution strategy (pre-selection of fleetsize *m*) given in Section 4.

(2) 2-Stage Heur is solved using our two-stage heuristic method with the four-overlap-breaker local search operator in the ALNS improvement process.

(3) 2-Stage ALNS is solved using our proposed two-stage heuristic method without the four-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.

(4) "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.

we further increase the model's size to 100 customers, none of the exact MIP approaches can terminate within hours, but our two-stage heuristic can still obtain results within 5 minutes, and within 10 minutes for the 150-customer instances.

For the solution quality, 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 the exact methods and two-stage heuristic, we observe that within the given time limit, our 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.

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.

To conclude, we are able to include last-minute customer cancellations into our solution process and make initial decisions based on probabilistic customer cancellations, 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. 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.

7 Summary

This paper studied the H-SARA problem, which integrates the fleet-sizing, assignment, routing, and scheduling problems. We have proposed a stochastic MIP model for the H-SARA problem, whose deterministic and stochastic versions are solved with two accelerated methods for small and medium scaled 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 two-stage 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 customer cancellation rates. Using our decision support framework, we can provide time and cost-effective decisions with low idling, waiting, and overtime costs, as well as two sets of customer appointment time windows, and balanced service team workload within geographically clear service zones.

References

- 1 Jonathan F Bard and Ahmad I Jarrah. Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B*, 43(5):542–561, 2009. doi:10.1016/j.trb.2008.10.003.
- 2 Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- 3 Dimitris J Bertsimas, Patrick Jaillet, and Amedeo R Odoni. A priori optimization. *Operations Research*, 38(6):1019–1033, 1990. doi:10.1287/opre.38.6.1019.
- 4 Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuysse. The vehicle routing problem: State of the art classification and review. *Computers & industrial engineering*, 99:300–313, 2016. doi:10.1016/j.cie.2015.12.007.
- 5 Paola Cappanera, Maria Grazia Scutellà, Federico Nervi, and Laura Galli. Demand uncertainty in robust home care optimization. *Omega*, 80:95–110, 2018. doi:10.1016/j.omega.2017.08.012.
- 6 Mohamed Cissé, Semih Yalçındağ, Yannick Kergosien, Evren Şahin, Christophe Lenté, and Andrea Matta. Or problems related to home health care: A review of relevant routing and scheduling problems. *Operations research for health care*, 13-14:1–22, 2017. doi:10.1016/j.orhc.2017.06.001.
- 7 George B Dantzig, Delbert R Fulkerson, and Selmer M Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi:10.1287/opre.2.4.393.
- 8 Chris Dawson. Royal mail day before delivery time notifications launched. URL: <https://tamebay.com/2019/04/royal-mail-day-before-delivery-time-notifications-launched.html>, April 2019.
- 9 DPD. Guide to dpd. https://www.dpd.co.uk/pdf/dpd_sales_guide_2020_v3.pdf, 2020.
- 10 Christian Fikar and Patrick Hirsch. A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105:300–310, 2015. doi:10.1016/j.jclepro.2014.07.013.
- 11 Christian Fikar and Patrick Hirsch. Home health care routing and scheduling: A review. *Computers & Operations Research*, 77:86–95, 2017. doi:10.1016/j.cor.2016.07.019.
- 12 The National Association for Home Care & Hospice. Basic statistics about home care, 2010. URL: http://www.nahc.org/wp-content/uploads/2017/10/10hc_stats.pdf.
- 13 Bruce Golden, Arjang Assad, Larry Levy, and Filip Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984. doi:10.1016/0305-0548(84)90007-8.
- 14 Diwakar Gupta and Brian Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819, 2008.
- 15 Elena Valentina Gutiérrez and Carlos Julio Vidal. Home health care logistics management problems: A critical review of models and methods. *Revista Facultad de Ingeniería Universidad de Antioquia*, 68:160–175, 2013.
- 16 Sandra Gutiérrez, Andrés Miniguano-Trujillo, Diego Recalde, Luis M Torres, and Ramiro Torres. The integrated vehicle and pollster routing problem. *arXiv*, 2019. arXiv:1912.07356.
- 17 Shuihua Han, Ling Zhao, Kui Chen, Zong-wei Luo, and Deepa Mishra. Appointment scheduling and routing optimization of attended home delivery system with random customer behavior. *European Journal of Operational Research*, 262(3):966–980, 2017. doi:10.1016/j.ejor.2017.03.060.
- 18 Vera C Hemmelmayr, Jean-François Cordeau, and Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & operations research*, 39(12):3215–3228, 2012. doi:10.1016/j.cor.2012.04.007.
- 19 S. W Hess, J. B Weaver, H. J Siegfeldt, J. N Whelan, and P. A Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006, 1965. doi:10.1287/opre.13.6.998.

- 20 Solrun G Holm and Ragnhild O Angelsen. A descriptive retrospective study of time consumption in home care services: How do employees use their working time? *BMC Health Services Research*, 14(1):439–439, 2014. doi:10.1186/1472-6963-14-439.
- 21 Simge Küçükyavuz and Suvrajeet Sen. An introduction to two-stage stochastic mixed-integer programming. In *Leading Developments from INFORMS Communities*, pages 1–27. INFORMS, 2017. doi:10.1287/educ.2017.0171.
- 22 Ettore Lanzarone and Andrea Matta. A cost assignment policy for home care patients. *Flexible Services and Manufacturing Journal*, 24(4):465–495, November 2011. doi:10.1007/s10696-011-9121-4.
- 23 Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009. doi:10.1287/trsc.1090.0301.
- 24 Canhong Lin, K.L Choy, G.T.S Ho, S.H Chung, and H.Y Lam. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4):1118–1138, 2014. doi:10.1016/j.eswa.2013.07.107.
- 25 Ran Liu, Biao Yuan, and Zhibin Jiang. A branch-and-price algorithm for the home-caregiver scheduling and routing problem with stochastic travel and service times. *Flexible Services and Manufacturing Journal*, 31(4):989–1011, 2019. doi:10.1007/s10696-018-9328-8.
- 26 P.A Maya Duque, M Castro, Kenneth Sörensen, and P Goos. Home care service planning. the case of landelijke thuiszorg. *European journal of operational research*, 243(1):292–301, 2015. doi:10.1016/j.ejor.2014.11.008.
- 27 Jorge Oyola, Halvard Arntzen, and David L Woodruff. The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics*, 7(3):193–221, 2018. doi:10.1007/s13676-016-0100-5.
- 28 César Rego, Dorabela Gamboa, Fred Glover, and Colin Osterman. Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European journal of operational research*, 211(3):427–441, 2011. doi:10.1016/j.ejor.2010.09.010.
- 29 María I Restrepo, Louis-Martin Rousseau, and Jonathan Vallée. Home healthcare integrated staffing and scheduling. *Omega (Oxford)*, 95:102057–, 2020. doi:10.1016/j.omega.2019.03.015.
- 30 Carlos Rodriguez, Thierry Garaix, Xiaolan Xie, and Vincent Augusto. Staff dimensioning in homecare services with uncertain demands. *International Journal of Production Research*, 53(24):7396–7410, 2015. doi:10.1080/00207543.2015.1081427.
- 31 Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006. doi:10.1287/trsc.1050.0135.
- 32 Nikolaos V Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6-7):971–983, 2004. doi:10.1016/j.compchemeng.2003.09.017.
- 33 Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming – CP98*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-49481-2_30.
- 34 Karmel S. Shehadeh and Mohan Chiriki. 13th aimms-mopta optimization modeling competition. In *Modeling and Optimization: Theory and Applications (MOPTA)*, 2021. URL: <https://coral.ise.lehigh.edu/~mopta/competition>.
- 35 Yong Shi, Toufik Boudouh, Olivier Grunder, and Deyun Wang. Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. *Expert systems with applications*, 102:218–233, 2018. doi:10.1016/j.eswa.2018.02.025.
- 36 Kenneth Sörensen and Marc Sevaux. A practical approach for robust and flexible vehicle routing using metaheuristics and monte carlo sampling. *Journal of mathematical modelling and algorithms*, 8(4):387, 2009. doi:10.1007/s10852-009-9113-5.

- 37 Biao Yuan, Ran Liu, and Zhibin Jiang. A branch-and-price algorithm for the home health care scheduling and routing problem with stochastic service times and skill requirements. *International Journal of Production Research*, 53:7450–7464, 2015. doi:10.1080/00207543.2015.1082041.
- 38 Yang Zhan and Guohua Wan. Vehicle routing and appointment scheduling with team assignment for home services. *Computers & Operations Research*, 100:1–11, 2018. doi:10.1016/j.cor.2018.07.006.
- 39 Yang Zhan, Zizhuo Wang, and Guohua Wan. Home service routing and appointment scheduling with stochastic service times. *European Journal of Operational Research*, 288(1):98–110, 2021. doi:10.1016/j.ejor.2020.05.037.

A Appendix

A.1 Districting MIP Formulation

Let $\llbracket 1, n \rrbracket$, 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 $\mu = \frac{1}{m} \sum_{i \in \llbracket 1, n \rrbracket} \omega_i$. We denote $\omega_{\min} \leq 100$ and $\omega_{\max} \geq 100$ 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:

$$\min \sum_{j \in \llbracket 1, n \rrbracket} \sum_{i \in \llbracket 1, n \rrbracket} \omega_i d_{i,j}^2 y_{i,j} \quad (7a)$$

$$\sum_{j \in \llbracket 1, n \rrbracket} y_{i,j} = 1 \quad \forall i \in \llbracket 1, n \rrbracket \quad (7b)$$

$$\sum_{j \in \llbracket 1, n \rrbracket} y_{j,j} = m \quad (7c)$$

$$y_{i,j} \leq y_{j,j} \quad \forall j \in \llbracket 1, n \rrbracket \quad (7d)$$

$$\sum_{i \in \llbracket 1, n \rrbracket} \omega_i y_{i,j} \geq \frac{\omega_{\min}}{100} \mu \cdot y_{j,j} \quad \forall j \in \llbracket 1, n \rrbracket \quad (7e)$$

$$\sum_{i \in \llbracket 1, n \rrbracket} \omega_i y_{i,j} + 2d_{0j} \leq L \quad \forall j \in \llbracket 1, n \rrbracket \quad (7f)$$

$$y_{i,j} \in \{0, 1\} \quad \forall i, j \in \llbracket 1, n \rrbracket \quad (7g)$$

Constraints (7b) require every customer to be assigned to a district. Constraint (7c) requires exactly m districts to be formed. Constraints (7d) state that each formed district must have a center. Constraints (7e) define the minimal workload of any district. Constraint (7f) 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 (or other self-defined upper bound using ω_{\max}).

A.2 ALNS Improvement Heuristic

A.2.1 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 [31] to remove 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 moved together with the $(q - 1)$ nearest customers from their tours to the customer pool.
4. *Tour Removal*: randomly remove a single tour. 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 $q = 5$ from experimental results. Customers inside the customer pool will be re-inserted by a repair operator selected from below [18]:

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.

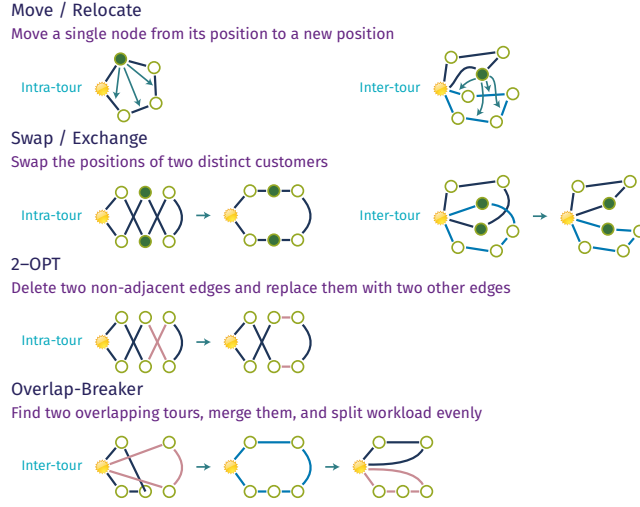
Since destroy and repair operators (with local search) allow us to modify the number of existing tours, it therefore is possible to re-optimize the fleet size during the ALNS search. As a result, ALNS allows our first-stage heuristic to be less affected by a poor selection of service team number m at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

A.2.2 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 5.

A.2.3 Roulette Wheel Selector with Adaptive Weight

We apply the *roulette wheel* (a probabilistic mechanism) to independently select the destroy and repair operators at each iteration. An operator i is selected with probability $\text{ow}_i / \sum_{k=1}^K \text{ow}_k$, where K is the group of same-category operators and ow_i is operator i 's



■ **Figure 5** Local Search Operators.

weight. The weight can be interpreted as the operator’s “capability” to bring improvement to the incumbent (best current solution), and can be mathematically updated to reflect its in-time performance in the previous N iterations as well as its overall performance throughout the ALNS solution process. We can compute operator i ’s weight in segment $j + 1$:

$$ow_{i,j+1} = \begin{cases} ow_{i,j}(1 - r) + \frac{\pi_i}{Q_i}r & \text{if } \pi_i > 0, \\ ow_{i,j} & \text{if } \pi_i = 0, \end{cases} \quad (8)$$

where $ow_{i,j}$ is the weight of operator i in the previous segment j . A segment is a consecutive number of iterations during the solution process. π_i is the score that operator i earned in segment j for contributing to improving the incumbent’s quality, and Q_i is the number of times operator i has been employed. Thus $\frac{\pi_i}{Q_i}$ is the average score operator i earns each time it was selected in segment j . This is weighted by a reaction factor r that controls how much the previous segment j determines each operator’s overall performance. Here we choose $r = 1/2$ based on experimental results.

A.2.4 Acceptance and Stopping Criteria

ALNS has an embedded *simulated annealing* (SA) meta-heuristic served as the acceptance criterion, which allows the algorithm to accept a newly-found solution s' that not necessarily brings a lower total cost. 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 $e^{f(s') - f(s) / T_{em}}$ where s is the current solution and T_{em} the initial temperature.

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

A.2.5 Further Improvements

To further improve on the real-life practicality of our routes and schedules derived after the districting-first-routing-second construction heuristic (Section 5.2) and ALNS improvement heuristic (A.2), 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 (7e) and (7f), 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 amongst the teams, we include a soft workload balance penalty $P \cdot \max\{|\frac{\sum_{i \in k} \omega_i - \mu}{\mu}| - \alpha, 0\}$ in the ALNS objective function to penalise the extra units of workload above or below a certain threshold α for any service team (district k) and an average workload μ amongst all districts. We have chosen $\alpha = 0.3$ 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 (since local search is computationally expensive). However, the application of *overlap-breaker* or *2-opt* (Figure 5) can remove the majority of overlaps and eliminate twisted tours that self-intersect.