Stochastic programming for an integrated assignment, routing, and scheduling problem

Syu-Ning Johnn Andrés Miniguano-Trujillo Yiran Zhu Akshay Gupte
Jörg Kalcsics

April 2, 2024

Abstract

We study a two-stage stochastic combinatorial optimisation problem that integrates fleet-sizing, assignment, routing, and scheduling problems. Although this problem has wide applicability, it arises in particular in the home healthcare industry where a service team of caregivers have to be assigned to patients and put in vehicle fleet that have to be routed amongst the patients they are going to serve, and one also needs to schedule within a single planning horizon and amidst uncertainty from factors such as service duration, travel time, and customer cancellation rates. A stochastic mixed-integer linear programming model is proposed. For the solution method, we first devise a Benders' decomposition algorithm with a heuristic-based warm-start. To accelerate the convergence process, we present a closed-form solution to the primal formulation for the subproblem to allow the computational time to increase linearly with the instance size. Furthermore, we provide a set of valid inequalities for the master problem that proved to be beneficial for reducing the number of feasibility cuts needed to be added and overall speeding up the convergence of the algorithm. In order to handle larger instances, we develop a two-stage heuristic to provide high-quality and timely solutions based on the information available at each stage. The proposed approach aims to reduce costs, create efficient routes with balanced workloads and team-based customer service zones, and increase customer satisfaction by implementing a two-stage appointment notification system that is updated at different stages before the actual service. The results from the computational experiments indicate that the proposed two-stage heuristic is highly effective in addressing our problem, as it can provide good-quality solutions for reasonable-sized instances and efficiently handle the uncertainty present in the problem. In particular, the heuristic is competitive with CPLEX's exact solution methods in terms of providing time and cost-effective decisions and can update previously-made decisions based on an increased level of information.

1 Introduction

We study a two-stage stochastic problem which integrates fleet-sizing, assignment, routing, and scheduling problems. This problem setting has a wide-range of real-world applications, particularly in the home service industry which encompasses businesses that provide a range of services to individuals in their homes. We motivate our problem by focusing on the home health care (HHC) sector, also known as domiciliary care or hospital-at-home services, which is an important part of home services. This type of care involves providing medical, paramedical, and social services to patients in out-of-hospital settings such as private homes, care facilities, boarding

School of Mathematics, University of Edinburgh, United Kingdom

homes, hospices, and shelters [LMS10]. In recent years, the HHC sector has experienced significant growth due to various societal changes, including shifts in family structures, increased life expectancy, an ageing population, hospital overcrowding, and the spread of infectious diseases.

The typical business requirements for HHC organisations are to decide 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 appointments to all customers with service demands. These four decisions form the Home Service Assignment, Routing, and Appointment scheduling (H-SARA) problem, which was presented for the 13th AIMMS-MOPTA Optimization Modeling Competition [SC21] [1]. H-SARA is a multifaceted optimisation problem that integrates planning decisions at tactical and operational levels and is related to a set of widely studied problems in academia and industry. It considers real-world complexities and uncertainties, such as travel times and service durations, making it a crucial problem to be addressed by HHC organisations.

Uncertainty and perturbation are inevitable in the healthcare system and cannot be neglected in real-life situations, where a planned schedule is often unlikely to be conducted throughout the planning horizon without any disruption. Amongst all uncertainties, variability in travel times, service times, and patients' availabilities are the most often appearing real-world unpredictabilities and have been widely studied in practice [DME21]. As a result, involving some uncertain aspects from real life within the healthcare decision framework is essential in producing practical solutions. The H-SARA problem considers the following real-life aspects: First, travel times can vary due to traffic congestion, potentially resulting in delays, especially during peak hours. Second, the actual service duration at each patient's location is intrinsically random. Third, patient presence during the decision-planning stage can be uncertain. For instance, patients can cancel at short notice after the healthcare provider has already made the routing and scheduling decisions, leading to abrupt modifications of service plans and timetables. We identified three customer cancellation scenarios on the service day and their associated policies: in-time cancellation, last-minute cancellation, and no-presence, all of which are further explained in Section 5.2. All three uncertainties can result in service teams incurring idle time or overtime, or patients experiencing waiting times.

The above uncertainties encompass both continuous (travel and service times) and discrete types (customer cancellations). Given their distinct nature, we assume in the following that customer cancellations are revealed right before the planning, whereas travel and service time uncertainties are intrinsic to the planning. To account for the uncertainty of the former, we delineate the problem into two distinct stages following a typical chronological timeline, the initial planning stage and the tour refinement stage, plus a post-service evaluation step, as depicted in Figure 1. This temporal progression facilitates a seamless daily operations rundown. The initial planning stage is carried out on the afternoon of the day prior to the actual service day. The planning uses as input all currently scheduled customer visits for the following day, i.e., all customer cancellations are assumed to have been received, plus the uncertain travel and service times. As output it determines the number of service teams to be used and their routes as well as an appointment time window for each customer of a certain length, e.g., 4 hours, which is then communicated to the customers.

In the morning of the actual service day, the tour refinement stage is carried out by a certain cut-off time and before the teams head out on their tours, considering all further customer cancellations received by then and also possible updates to travel and service times (any customer

[1] This paper is an extension of the the proceedings paper [Joh+21] which contained the authors' submission to the AIMMS-MOPTA Modeling Competition 2021 at which they were awarded the First Prize.

cancellations received after the cut-off time will have to be addressed ad-hoc by each team). The tour refinement stage allows re-optimizing the number of service teams, the team-customer allocations and the service routes in order to better balance workload and minimize overtime and waiting and idling times. However, it will generate and communicate narrower appointment time windows to customers, e.g., of 30 minutes length, which should be nestled within the time windows that was communicated to customers on the previous day. To ensure the robustness of the solution of the tour refinement stage, an out-of-sample simulation can be run that provides feedback to the planner.

To complete the planning process, a post-service evaluation step takes place at the end of the service day, allowing decision-makers to evaluate the service teams' workload and, e.g., compensate teams which ran into overtime by assigning them fewer customers on the next day.

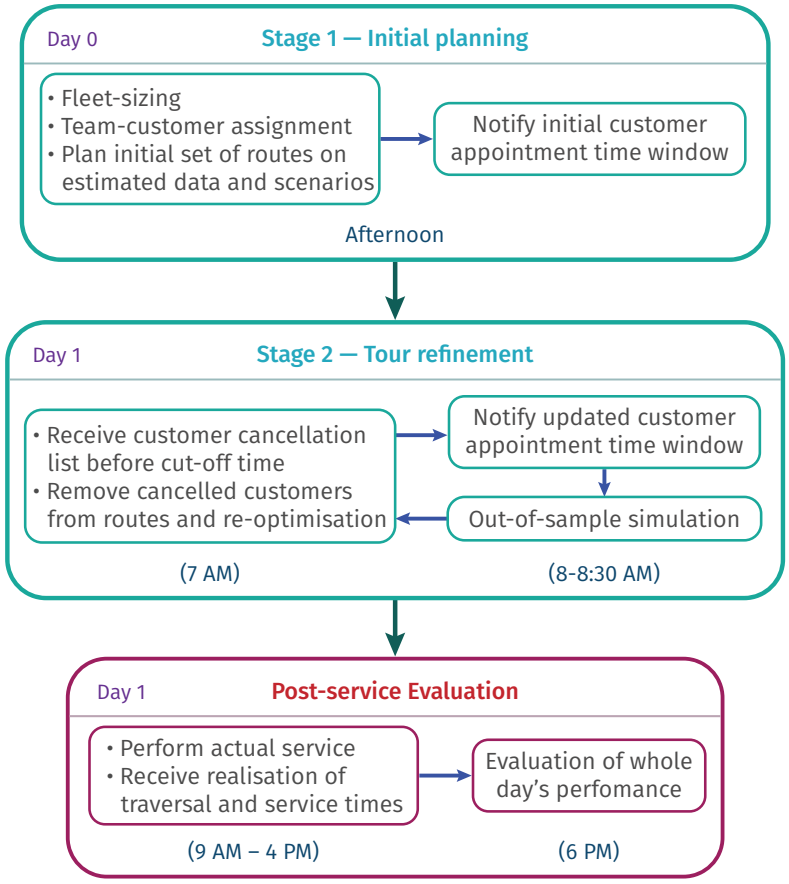


Figure 1: Daily operations rundown with chronological timeline. Optimisation steps are marked in green boxes and evaluation steps are marked in red boxes.

To solve the models for the initial planning and tour refinement stage, we propose an exact method and a heuristic. With respect to the three types of uncertainties, both methods consider travel and service time uncertainties included via a finite set of scenarios, but assume the list of customers to be served to be known in advance.

Concerning the exact method, we first derive a stochastic mixed-integer linear programming (MIP) model for H-SARA and then propose a *Benders' decomposition* (BD) for solving the deterministic equivalent. Realising the natural partition of our stochastic model, the master problem decides on the number of teams, their routes, and the appointment time windows, whereas

the subproblems compute the actual scenario-dependent arrival times of teams at customers and the incurred waiting and idling times as well as overtimes. The Benders' decomposition presented here is a significantly improved version of the one reported in Johnn et al. [Joh+21]. To be able to solver larger instances in reasonable time, especially for the tour refinement stage, we also develop an *adaptive large neighbourhood search* (ALNS) meta-heuristic that provides high-quality and in-time solutions.

Our computational results show that our Benders' decomposition can deliver good-quality solutions with better efficiency than the MIP model solved by CPLEX. Our BD is also competitive with the CPLEX built-in BD for small-scale instances. Moreover, our tailored ALNS heuristic is competitive to CPLEX's exact solution methods in providing time and cost-effective decisions for large-scale instances.

The organisation of this paper is as follows. In Section 2, we review the studies related to human resource planning in HHC. In Section 3 we define the general problem (which subsumes the H-SARA problem) and formulate the problem as a two-stage stochastic MIP model. To solve the problem, we present the Benders' decomposition in Section 4 and a tailored two-stage heuristic solution method in Section 5. Computational results and performance analysis are presented in Section 6. Conclusion follows in Section 7.

2 Literature Review

Our problem consists of determining the number of required agents (in the context of H-SARA these would be caregivers) and their visiting routes to minimise the total hiring, service, and travel expenses while satisfying a set of constraints. It relates to a set of widely studied academic and industry problems. In this section, we summarise several relevant problems in the context of HHC.

2.1 Related Problems

The *vehicle routing problem* (VRP) is a significant issue at an operational level. The VRP has been extensively studied in the literature and has several variants and applications [Lap09; Lin+14; BRV16]. The problem studied in this paper generalises the VRP to determine the necessary personnel to meet demand at the lowest cost. This approach includes a fleet-sizing aspect, commonly known as the staffing or personnel planning problem, to balance fixed team hiring costs and routing expenses [Rod+15]. Many HHC problems also involve a patient-to-staff or patient-to-slot assignment aspect over a planning horizon [CLM18], which is considered alongside other planning decisions.

Another related operational decision is *scheduling*, which refers to the chronological allocation of tasks to workers such that the list of tasks is accomplished within the shortest amount of time and with minimal time clashes. For example, in healthcare, *appointment scheduling* specifies the allocation of patients to staff for services, including care visits and elective surgeries. It aims at constructing staff schedules that satisfy patient requirements. From a service provider's perspective, each customer visit is scheduled as part of a service team's timetable in sequential order [GD08]. The *vehicle routing problem with time windows* (VRP-TW) [DMR14] is a VRP variant with a scheduling aspect, stressing that vehicle arrival or departure times must satisfy additional customer availability requirements. The difference between scheduling and the VRP-TW is the pro-activeness of the decision-makers in appointing the visiting time window due to the initial routing criteria instead of customer-imposed time requirements.

At an operational level, our problem can be formulated as the *routing and scheduling problem* (RSP), which involves producing for every caregiver a job schedule and a visiting sequence at minimal cost respecting regulatory and operational constraints [DME21]. RSP in the HHC context has received growing attention and has been the predominant research focus [ND18]. For comprehensive reviews of RSP in the context of HHC, we refer the readers to Fikar and Hirsch [FH17] with a focus on the standard parameters, Cissé et al. [Cis+17] with analysis on Operations Research models applied to RSP, and Di Mascolo et al. [DME21] with specification on the RSP constraints and objectives in RSP. Besides, Gutiérrez and Vidal [GV13] review the main models and decisions related to the design and delivery of HHC services. Furthermore, a more recent survey by Grieco et al. [GUC20] gives a systematic review of a broader framework of Operations Research literature in HHC logistics management. Finally, a comprehensive overview of planning decisions in health care can be found in Hulshof et al. [Hul+12].

2.2 Uncertainties

There is a growing trend in the recent HHC literature that deals with uncertainties such as travel times, service times variability, or customer cancellations. Yuan et al. [YLJ15], Shi et al. [SBG17], Zhan and Wan [ZW18], and Shi et al. [SBG19] consider uncertain service times or demand for their RSP models. Rest and Hirsch [RH16] consider time-dependent travelling time for the staff routing and rostering problem. Several works involve customer cancellation or departure in a routing and scheduling context. For instance, in the work of Eveborn et al. [EFR06] and Eveborn et al. [Eve+09], the authors consider changes in staff and patients' availabilities in their staff routing and scheduling problem in home care. Trautsamwieser et al. [TGH11] considers personal preferences from both patients and nurses that allow refusal of service. Nickel et al. [NSS12], Fikar et al. [Fik+16], and Cappanera et al. [Cap+18] and Gomes and Ramos [GR19] all consider the uncertainty of patient inflow due to patient cancellation or new requests. Finally, Lin et al. [Lin+18] considers sudden incidents, which include nurse leave, patients requesting changes of their assigned time slot, and patient cancellation. Furthermore, some other complex models include more than one type of the above uncertainties in the literature. Works from Shi et al. [Shi+18], Yuan et al. [YLJ18], Liu et al. [LYJ19], Nikzad et al. [NBA21], Shahnejat-Bushehri et al. [Sha+21], and Naderi et al. [Nad+23] and Yang et al. [YNY21] consider travel and service times uncertainties. Han et al. [Han+17] considers uncertainties in staff response time and customer cancellation in the form of no-shows. A review of other types of uncertainties apart from the above three can be found in the work of Di Mascolo et al. [DME21]. Nevertheless, to our best knowledge, no research integrates all three types of uncertainties together with the fleetsizing, assignment, routing, and appointment scheduling decisions in the context of HHC.

Many solution methods have been applied to handle uncertainty, amongst which stochastic programming has been one of the most selected methods that shows good performance in the HHC context [HPR20; DME21; ZWW21]. A common approach applied in stochastic programming models is two-stage optimisation. In the first stage, an initial solution is created before the cancellation parameters are revealed in the second stage, meaning that first-stage decisions should possess sufficient flexibility to guarantee the feasibility of second-stage recourse actions. Although this paper focuses on a problem emerging in home service delivery, decomposition methods, specifically two-stage optimisation, can be easily found in practice. 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 deliv-

ery when more information is known (e.g., customer delivery sequence, cancellations). This multi-stage decomposition approach also suits the real-life circumstances in the HHC service industry, where last-minute service cancellations, i.e. customers cancelling their requests after being given an appointment time, are allowed.

The use of decomposition techniques, specifically Benders' decomposition, to solve integrated large-scale NP-hard problems like ours has gained considerable significance in recent years [Rah+17; Nad+23]. One specific application of Benders' decomposition is the L-shaped method, which is used to solve stochastic problems. The method has been shown to be particularly effective in solving these kinds of problems and has therefore positioned itself in the field. The L-shaped method breaks the problem down into smaller sub-problems, which can then be solved separately and combined using cuts to find the overall solution. This approach can effectively tackle complex problems that would be difficult to solve using traditional methods. As a result, we believe the L-shaped method is a good fit for our problem settings and could be an effective solution for similar service problems.

Table A.1 gives an extensive summary of the literature on HHC with stochasticity and focuses on routing and scheduling. The most similar works to ours in terms of solution methods are Hashemi Doulabi et al. [HPR20] and Zhan et al. [ZWW21], both of which considered stochasticity in their models. The former studied the stochastic VRP with synchronised staff visits and uncertain travel and service times and proposed an L-shaped algorithm and a branch-and-cut implementation as solution methods for instances up to 20 customers. In contrast, the latter studied an integrated routing and scheduling problem with stochastic service times and proposed an L-shaped algorithm for instances up to 10 customers and a problem-specific heuristic for larger-sized instances.

2.3 Our Novelties and Contributions

Based on our literature review, the main contribution of this work is the mathematical treatment of a problem integrating the following four planning decisions: fleet-sizing, assignment, routing, and scheduling. These decisions are jointly integrated into the solution framework to contribute to a comprehensive decision-making process. 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. For the solution methods, we first develop an exact method using Benders' decomposition based on the natural partition of our two-stage stochastic model to tackle small-sized instances. We propose a set of valid inequalities for the master problem that proved to be beneficial for reducing the number of feasibility cuts needed to be added and overall speeding up the convergence of the algorithm. Besides, we present a closed-form primal formulation for the subproblem to allow the computational time to increase linearly with the instance size. Moreover, we develop a metaheuristic-based warm-start algorithm to accelerate the convergence process. An alternative two-stage heuristic approach is proposed for tackling medium-to-large sized instances, allowing decision-making based on imperfect information before the actual customer demands are revealed and updating existing solutions with increased information.

3 Mixed Integer Programming Model

3.1 Problem Statement and Parameters

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 ordered pairs linking each pair of customers, one link from the depot to each customer and another from each customer back to the duplicated depot; 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 an optimal 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 time slots, either on the day before after the initial planning or 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.

Concerning the uncertainties, we assume that all cancellations are known by the time of planning, whereas the travel and service times remain uncertain, but with known probability distributions and the associated variables are assumed to be independent. The set Ω denotes different scenarios ω, each associated with a different realisation of the travel and service times with a certain probability q_ω. We assume that Ω is a finite set and denote its cardinality by $|\Omega|$. We impose a traversal duration matrix $T := (\tau^\omega_{i,j})$ under scenario ω for any arc $(i, j) \in A$ based on the stochastic traversal variables, and a service duration vector $S := (s^\omega_i)$ for customer i under scenario ω from the stochastic service variables. The distance from i to j is labelled $d_{i,j}$. In this formulation, symmetry of T and d is not required, capturing possible discrepancies in the underlying road network, i.e., city topography and street layout. Let $p : [\![1, n]\!] \to [0, 1]$ be a probability mass function defined over the set of customers, such that for each customer $i \in [\![1, n]\!]$ the probability of service cancellation of i is given by p_i. The cost of hiring a homogeneous team $i \in [\![1, m]\!]$ is taken as a constant f_m, where m is the maximum number of service teams available. The regular allowed working time per day is given by $L \geq 0$. Working times are expected to be within the interval $[0, L]$, yet we anticipate and allow overtime by an amount of up to $\theta > 0$. Any additional time beyond the regular working time L and within $L + \theta$ results in an overtime cost. Finally, let c_{wait}, c_{idle}, and c_{over} be the fixed non-negative unit customer waiting, team idling, and team overtime costs, respectively.

3.2 Stochastic MIP Model

In the following we present a stochastic model and its deterministic equivalent for the H-SARA problem.

3.2.1 Two-stage MIP Formulation

We decompose the problem into two interconnected stages based on scenario dependency. All decision variables relating to fleet size, customer-team allocation, team routing, and customer appointment time windows are scenario-independent and are kept in the first stage, whereas variables computing the actual arrival times as well as idling and waiting times are scenario-dependent are included in the second stage.

Fleet size, customer-team allocation, and team routing decisions are modelled through a single binary traversal variable $x_{i,j} \in \{0,1\}$ that equals to 1 if and only if arc $(i,j) \in A$ is traversed by a service team. Notice that the fleet size can be derived using $x_{i,j}$ if we consider the total number of edges linking customers to the depot is exactly twice the size of the fleet, and that $x_{i,i}$ due to the domain restriction is not allowed in the model. Since the actual arrival time of a team under the stochastic setting could be different from the predicted one, we introduce a scenario-independent variable t_i that computes the expected arrival time at customer i. We then assume that the appointment time window is centred at t_i with a fixed width of $2W$, $W > 0$, i.e., $[t_i - W, t_i + W]$. The actual arrival of a service team before the scheduled appointment time window leads to team idling, whereas an arrival after the time window leads to the customer waiting. This is modelled in the second stage where we employ a set of continuous variables for each customer $i \in [\![1,n]\!]$ and each scenario $\omega \in \Omega$: $a_i^\omega \in [0,L]$ and $h_i^\omega \geq 0$ for the team's arrival and idling time at i, respectively; $w_i^\omega \geq 0$ for the customer's waiting time; and g_i^ω for the overtime of a service team registered at their arrival at the depot when returning from the last customer i in the tour.

The first-stage formulation for the stochastic model is then as follows:

$$(1^{\text{st}}\text{ Stage}) \qquad \min_{(x;t)} \quad f_m \sum_{i\in[\![1,n]\!]} x_{i,n+1} + \sum_{(i,j)\in A} d_{i,j} x_{i,j} + \mathbb{E}\left[Q(x,t;\omega)\right] \tag{1a}$$

subject to

$$\sum_{i\in[\![1,n]\!]} x_{0,i} \leq m \tag{1b}$$

$$\sum_{\substack{i\in[\![0,n]\!]\\ i\neq j}} x_{i,j} = 1 \qquad j \in [\![1,n]\!] \tag{1c}$$

$$\sum_{\substack{i\in[\![0,n]\!]\\ i\neq j}} x_{i,j} = \sum_{i\in[\![1,n+1]\!]} x_{j,i} \qquad j \in [\![1,n]\!] \tag{1d}$$

$$\sum_{i\in[\![1,n]\!]} x_{0,i} = \sum_{i\in[\![1,n]\!]} x_{i,n+1} \tag{1e}$$

$$\widehat{s}_i + \widehat{\tau}_{i,j} \leq L(1 - x_{i,j}) + t_j - t_i \qquad i,j \in [\![1,n]\!], i \neq j \tag{1f}$$

$$0 \leq t_i + \widehat{s}_i + \widehat{\tau}_{i,0} \leq L \qquad i \in [\![1,n]\!] \tag{1g}$$

$$x_{i,j} \in \{0,1\} \qquad (i,j) \in A \tag{1h}$$

The objective function (1a) minimises the total team hiring cost, routing cost, and the expected second stage cost $\mathbb{E}\left[Q(x,t;\omega)\right] = \sum_{\omega\in\Omega} q^\omega \cdot Q(x,t;\omega)$ for idling, waiting, and overtime for any x and t satisfying the above equations and over all scenarios $\omega \in \Omega$ associated with probability q^ω. Constraint (1b) states that there are no more than m homogeneous service teams departing from the depot $\{0\}$. Constraints (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. Constraints (1f) link the traversal variable $x_{i,j}$ to the expected customer arrival time t_i using the the expected travel time $\widehat{\tau}_{i,j}$ and the expected service time $\widehat{s}_i$. Constraints (1g) ensure that, on expectation, a service team can complete its tour within the regular working time L

(we only allow overtime in the second stage). Additionally, (1f) are a variant of the Miller-Tucker-Zemlin (MTZ) subtour elimination constraints that rule out any cycles not containing the depot. Lastly, (1h) are the domain constraints for the traversal variable.

The second-stage formulation for the stochastic model is as follows:

$$(2^{\text{nd}}\text{ Stage}) \qquad Q(x,t;\omega) := \min_{w,h,g} \quad c_{\text{wait}} \sum_{i\in[\![1,n]\!]} w_i^\omega + c_{\text{idle}} \sum_{i\in[\![1,n]\!]} h_i^\omega + c_{\text{over}} \sum_{i\in[\![1,n]\!]} g_i^\omega \tag{2a}$$

subject to

$$a_i^\omega + h_i^\omega + s_i^\omega + \tau_{i,j}^\omega \le a_j^\omega + M(1 - x_{i,j}) \qquad \forall (i,j) \in A, \tag{2b}$$
$$a_i^\omega + h_i^\omega + s_i^\omega + \tau_{i,j}^\omega \ge a_j^\omega - M(1 - x_{i,j}) \qquad \forall (i,j) \in A, \tag{2c}$$
$$a_i^\omega + h_i^\omega + s_i^\omega + \tau_{i,n+1}^\omega \le L + g_i^\omega + \theta(1 - x_{i,n+1}) \qquad \forall i \in [\![1,n]\!], \tag{2d}$$
$$h_i^\omega \ge (t_i - W) - a_i^\omega \qquad \forall i \in [\![1,n]\!], \tag{2e}$$
$$w_i^\omega \ge a_i^\omega - (t_i + W) \qquad \forall i \in [\![1,n]\!], \tag{2f}$$
$$g_i^\omega \le \theta \qquad \forall i \in [\![1,n]\!], \tag{2g}$$
$$a_i^\omega, h_i^\omega, w_i^\omega, g_i^\omega \ge 0 \qquad \forall i \in [\![1,n]\!], \tag{2h}$$
$$a_0^\omega = h_0^\omega = 0 \qquad \forall i \in [\![1,n]\!]. \tag{2i}$$

where the scenario-based objective function (2a) minimises the idling, waiting, and overtime costs under scenario ω. Constraints (2b) and (2c) 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. Hereby, M is a large number in these constraints, e.g., $L + \theta$ would be sufficient. Constraints (2d) determine the incurred overtime when returning to the depot from the last customer. Constraints (2e) and (2f) specify the idling and waiting times, respectively. Constraints (2g) restrict the extent of overtime permissible in the second stage due to uncertainties related to travel and service. Together with (2d) they make sure that, any second-stage tour must not exceed a total time length of $L + \theta$. Constraints (2h) provide domain restrictions for the relevant variables. Finally, an additional requirement of no arrival or idle times at the depot must be enforced by (2i).

4 L-shaped Method

Benders' decomposition [Ben62] is a frequently-applied technique for solving large-scale MIP problems with a block-diagonal structure. It is called the *L-shaped method* when applied to two-stage problems. Benders' decomposition has a "row-generation" strategy that relaxes the original problem and iteratively generates cuts to strengthen the relaxed problem until convergence is achieved.

4.1 Benders Algorithm

We introduce the Benders's algorithm below. A given stochastic programming model containing both global and local constraints can be represented as $z_{\text{MIP}} := \min \left\{ cx + \sum_{\omega\in\Omega} dy^\omega : Ax + By^\omega \ge b,\, Dx \ge f,\, x \in \mathbb{Z}^n,\, y^\omega \in \mathbb{R}^m,\, \omega \in \Omega \right\}$, Benders' decomposition separates the complex optimisation problem into the master problem (MP) and subproblem. It keeps the complicated linking variable x within the MP, which can be written as $z_{\text{MP}} := \min \left\{ cx + z_{\text{SP}}(x) : Dx \ge f,\, x \in \mathbb{Z}^n \right\}$ and projects out the remaining variables in a composite format which yields the

subproblem's objective value as the sum of different scenarios $z_{\text{SP}} = \sum_{\omega \in \Omega} z^{\omega}_{\text{SP}}$. Here, for a fixed first-stage decision vector $x = \overline{x}$, the remaining subproblem can be formulated as $z^{\omega}_{\text{SP}}(\overline{x}) := \min \left\{ dy^{\omega} : By^{\omega} \geq b - A\overline{x},\, y^{\omega} \in \mathbb{R}^m \right\}$ which can be solved independently for each scenario $\omega \in \Omega$. Since the second-stage variable y^{ω} is continuous for any scenario ω, each subproblem is a linear programming problem and therefore has a corresponding dual formulation $z^{\omega}_{\text{DP}}(\overline{x}) := \max \left\{ u(b - A\overline{x}) : ub \leq d,\, u \geq 0 \right\}$, where u is the vector of dual multipliers corresponding to the constraints $By^{\omega} \geq b - A\overline{x}$. The polyhedron of the dual formulation can be represented in terms of its extreme points and extreme rays. We begin by solving the MP. For each iteration of the Benders algorithm, a vector $\overline{x}$ can be derived from the reduced master problem (RMP). The subproblems are linked to the RMP through two types of additional cuts. In the first case, if the corresponding dual subproblem $z^{\omega}_{\text{DP}}(\overline{x})$ has a bounded and nonempty feasibility region, a Benders optimality cut $u(b - Ax) \leq \varphi$ is generated using the extreme point u, which are the corner points of the feasible region that satisfy a particular subset of constraints. φ is a scalar value determined by the SP solution that represents the threshold for the violation of constraints at the extreme point that is allowed to satisfy optimality. The generated optimality cuts are added to the RMP for the consecutive iteration. Otherwise, if the subproblem dual is unbounded and determines that the first-stage realisation $\overline{x}$ is infeasible, a Benders feasibility cut $u(b - Ax) \leq 0$ is generated using the extreme ray u and added to the RMP.

Compared to the single-cut Benders' decomposition that generates a single optimality cut integrating all the subproblems for each iteration, the multi-cut reformulation produces a cut from each subproblem, which can lead to a more aggressive pruning of the solution space and therefore be potentially more efficient compared to the classical single-cut method. The RMP is iteratively updated and solved to generate new Benders cuts until the gap between the upper and lower bounds on the objective function value falls below the termination threshold.

4.2 Benders Framework Overview

We visualise our customised Benders' decomposition process in a flowchart depicted in Figure 2. The process commences with the upper centre box in red on the flowchart, where we first relax the MIP formulation for our stochastic programming model by separating the first-stage traversal decision from the time-related decisions that are scenario-dependent in the second stage. We form the initial RMP based on constraints (1a)–(1f), solving the resulting MIP model yields a set of visiting sequences x and appointment times t for each customer. A preprocessing warm start can be included prior to solving the RMP and this is also explained as part of the MP in Section 4.3. After validating the tours for feasibility and eliminating any illegal subtours that involve sequences without visiting the depot, we proceed to derive the master solution and record the lower bound. At the MP, any detected subtour formed by a group of customer nodes N_s without the inclusion of a depot is used to generated subtour elimination constraints and added to the RMP. Afterwards, we determine the second-stage time-related variables a, g, h, w for the scenario-based subproblems (2a)–(2h), which are used to verify the feasibility of the master-level traversal decision and generate the corresponding Benders cuts. At this stage, we check for any group of nodes N_D causing overtime infeasibility, which are again used to generate constraints that are added to the RMP. To derive solutions for a large number of subproblems more efficiently, we develop a closed-form formulation explained in Section 4.4 for each primal subproblem with realisations of scenario-dependent travel and service time. The generations of Benders' optimality and feasibility cuts are explained in Section 4.5. Specifically, if the master-level traversal decision is feasible for a certain scenario, a Benders optimality cut is generated using the dual multipliers from the subproblem. For any infeasible scenario,

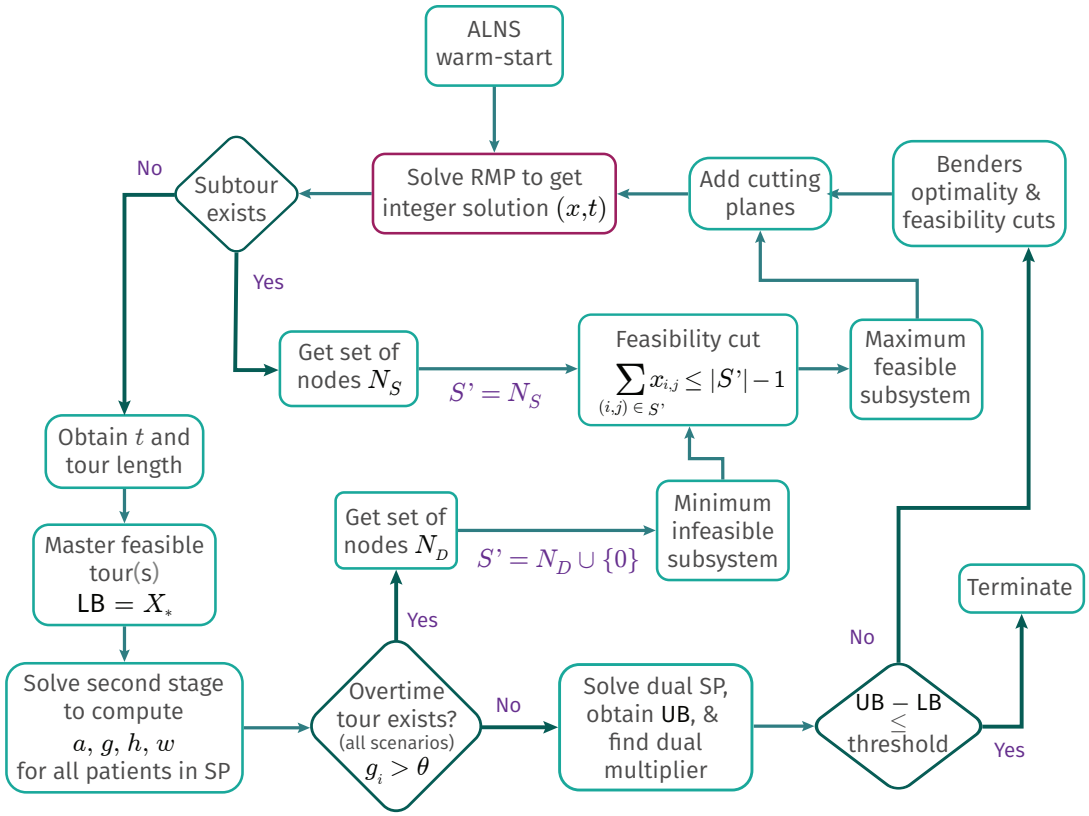


Figure 2: Benders' decomposition flowchart

we generate a feasibility cut to forbid the formation of an overtime tour visiting an oversized group of customers. While the first-stage solution does not allow overtime, this is based on the expected travel and service times. Due to travel and service time uncertainty, for a specific scenario the actual working time of a team may not only lead to overtime, but even exceed $L+\theta$. To cope with the latter, we have introduced feasibility cuts in the form of a subtour elimination constraint. Furthermore, we provide some acceleration techniques using the minimum infeasible subsystem (MIS) and maximum feasible subsystem (MFS) for generating optimality cuts based on infeasible scenarios and strengthening feasibility cuts will be explained in Section 4.6 and Section 4.7, respectively. Finally, the iteration terminates when the MP lower bound and subproblem upper bound satisfy a predetermined threshold. Regarding the overall Benders method, we utilise a single tree approach, where the first stage is solved optimally as an MIP model. The generated feasibility and optimality cuts from the second stage are incorporated using the callback function to reintroduce them to the first stage model.

4.3 Master Problem

In order to limit the domain of our first-stage decision variables x and t, we have added the following valid inequalities to the MP to shrink the feasibility region:

$$t_i \geq \min_{\substack{\omega \in \Omega \\ j \in V}} \tau^{\omega}_{0,j} \qquad \forall i \in [\![1, n]\!], \tag{3a}$$

$$|t_i - t_j| \geq \min\{\tau^{\omega}_{i,j}, \tau^{\omega}_{j,i}\} + \min\{s^{\omega}_i, s^{\omega}_j\} \qquad \forall (i,j) \in A. \tag{3b}$$

where constraint (3a) requires that the appointment time for each customer cannot precede the completion of the team's journey from the depot in any scenario. Notice that the triangular inequality does not hold due to the stochasticity in travel time. Constraint (3b) states that the intermediate appointment time for any pair of customer nodes must be equal to or greater than the combined duration of travel and service times needed for either direction of visit.

4.3.1 Lifted Formulation for Subproblem

We further consider a lifted formulation to strengthen the MP. We start with the following lifted valid inequality. For simplicity, we do not explicitly show the scenario index for all second-stage variables.

$$a_j \geq \sum_{i \in [\![1,n]\!]} (a_i + h_i + s_i + \tau_{i,j}) x_{i,j} \tag{4}$$

The inequality above is derived after considering any two consecutive customers i and j in the subproblem solution. To see this, first observe that

$$a_j x_{i,j} = (a_i + h_i + s_i + \tau_{i,j}) x_{i,j} \tag{5}$$

holds for all arcs (i, j). Indeed, if $x_{i,j} = 0$, this is trivial, and thus let us focus on the case $x_{i,j} = 1$. Combining constraints (2b) and (2c) gives a linear constraint with equality in the form $a_i + h_i + s_i + \tau_{i,j} = a_j$, which is exactly (5). We then accumulate (5) over all start nodes $i \in [\![1, n]\!]$ with the same end node j. Since $\sum_{i \in [\![1,n]\!]} x_{i,j} \leq 1$ holds for all nodes j, multiplying it by a_j to obtain $a_j \geq \sum_{i \in [\![1,n]\!]} x_{i,j} a_j$. This allows us to replace the left hand side in the sum by a_j which results in (4). However, equation (4) exhibits a quadratic form as a result of the product involving the variables $a_i x_{i,j}$ and $h_i x_{i,j}$. To convert this equation into its linear equivalent, we use McCormick envelopes [McC76] through the introduction of auxiliary variables ς and ϱ. These auxiliary variables are characterised by the following set of inequalities:

$$\varsigma_{i,j} \leq x_{i,j}, \quad \varsigma_{i,j} \leq a_j + (L + \theta)(1 - x_{i,j}), \quad \varsigma_{i,j} \geq 0, \quad \varsigma_{i,j} \geq a_j - (L + \theta)(1 - x_{i,j}), \tag{6a}$$

$$\varrho_{i,j} \leq x_{i,j}, \quad \varrho_{i,j} \leq h_j + (L + \theta)(1 - x_{i,j}), \quad \varrho_{i,j} \geq 0, \quad \varrho_{i,j} \geq h_j - (L + \theta)(1 - x_{i,j}). \tag{6b}$$

It is worth noting that $L + \theta$ represents the maximum workload allowed, which also serves as an upper bound for a_j and h_j. The set of inequalities in (6) ensures that $\varsigma_{i,j} = a_j x_{i,j}$ and $\varrho_{i,j} = h_j x_{i,j}$ when $x_{i,j} \in \{0, 1\}$. Throughout the paper, we will employ the lifted formulation by incorporating (6) and substituting $a_j x_{i,j}$ and $h_j x_{i,j}$ with $\varsigma_{i,j}$ and $\varrho_{i,j}$, respectively.

4.3.2 Warm Start

To warm-start the MIP solver, we implement a metaheuristic based on an *adaptive large neighbourhood search* to install a feasible initial solution for the MP. The metaheuristic methodology is further introduced in Section 5.3.4.

To produce such an initial solution through a metaheuristic, we propose a *root-node method* to select a feasible fleet size while addressing a trade-off between fixed vehicle costs and routing costs. Since service team hiring costs typically dominate the remaining costs, starting the search process with a near-optimal fleet size is crucial to discovering the optimal solution and promising solution candidates in reasonable computational time.

The tightness of constraint (1b) is strongly dependent on the value of m, which is generally large given that the number of available service teams might be more than the number eventually hired. Therefore, we aim to provide a tighter restriction on m by establishing both upper and

lower bounds for the service teams, such that the domain is effectively constrained. The *root-node method* begins with limiting the fleet size options using the upper and lower bounds given below. For notational simplicity, the hat decorator in the travel and service times involved in the following constraints are the expected values for each arc and node, namely $\widehat{\tau}$ and $\widehat{s}$. We can find an upper bound on the number of teams required to visit all clients by solving the following linear programme:

$$\min \ell_u \tag{7a}$$

subject to

$$\sum_{i \in [\![1,n]\!]} \widehat{s}_i + \frac{1}{2} \sum_{i \in V} \left(\max\{\widehat{\tau}_{i,j} : (i,j) \in A\} + \max\{\widehat{\tau}_{j,i} : (j,i) \in A\} \right) \leq \ell_u (L + \theta), \tag{7b}$$

$$1 \leq \ell_u \leq \widehat{m}, \qquad \text{and} \qquad \ell_u \in \mathbb{Z}; \tag{7c}$$

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 service requirements. Here, $\widehat{m}$ 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 (7) determines a choice over m. Observe that if we divide constraint (7b) 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 to travel between customers taking time-consuming paths. Notice that the optimal solution can be obtained using exhaustive enumeration in $\mathcal{O}(1)$ 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:

$$\max \frac{1}{\ell_l} \tag{8a}$$

subject to

$$\sum_{i \in [\![1,n]\!]} \widehat{s}_i + \frac{1}{2} \sum_{i \in V} \left(\min\{\widehat{\tau}_{i,j} : (i,j) \in A\} + \min\{\widehat{\tau}_{j,i} : (j,i) \in A\} \right) \leq \ell_l (L + \theta), \tag{8b}$$

$$1 \leq \ell_l \leq \widehat{m}, \qquad \text{and} \qquad \ell_l \in \mathbb{Z}; \tag{8c}$$

Notice that if this problem is infeasible, then there are not enough teams to solve the problem with mean values for service and transportation times. As a result, we have an infeasibility certificate. Again, this problem can be solved in $\mathcal{O}(1)$ time.

After shrinking the fleetsize domain using the bounds above, we enumerate through the list of possible fleetsize options in $\{\ell_l, \ell_l + 1, \ldots, \ell_u\}$ and for each fixed fleetsize value, the CPLEX solver is instructed to stop at the first integer solution found in the root node before the branching begins. After all associated root node values are computed, we instruct CPLEX to identify the smallest root node value out of the list of integer solutions amongst all fleetsize options and return its associated fleetsize $\widehat{m}$, which is used to provide an initial solution during the warm-start. This way, we find a good starting point of the search with fleet size $\widehat{m}$ for the specific travel and service times realisations. As a result, the warm-start metaheuristic produces an initial solution with $\widehat{m}$ service tours for the RMP. We use the expected travel and service times from all scenarios during the warm start to generate such tours.

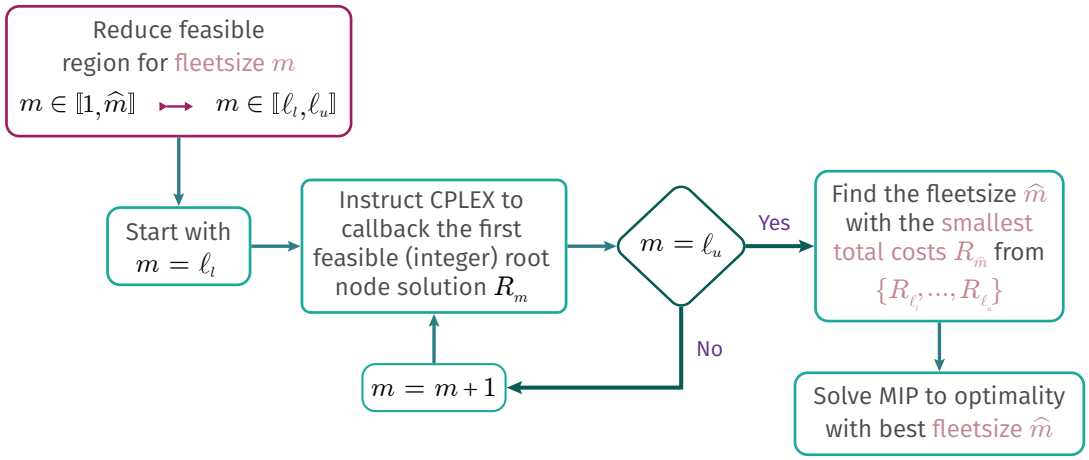


Figure 3: Depiction of warm-start

4.4 Primal Solution for SP

Once the first-stage routing plan is established, the second-stage becomes a linear programming problem, for each scenario $\omega \in \Omega$, resembling a network flow problem. In fact, as the routing variables x are fixed for this stage, constraints (2b) and (2c) can be simplified, significantly reducing the size of the model. More explicitly, and dropping the scenario index, the following reduced second stage model is obtained:

$$\text{(Reduced 2}^{\text{nd}}\text{ Stage)} \qquad Q_{\mathsf{R}}(x,t,\omega) := \min_{w,h,g} \; c_{\text{wait}} \sum_{i\in[\![1,n]\!]} w_i + c_{\text{idle}} \sum_{i\in[\![1,n]\!]} h_i + c_{\text{over}} \sum_{i\in[\![1,n]\!]} g_i \tag{9a}$$

subject to

$$a_i + h_i + s_i + \tau_{i,j} = a_j \qquad \forall (i,j) \in A : x_{i,j} = 1, j \neq n+1, \tag{9b}$$
$$a_i + h_i + s_i + \tau_{i,n+1} \leq g_i + L \qquad \forall i \in [\![1,n]\!] : x_{i,n+1} = 1, \tag{9c}$$
$$h_i \geq (t_i - W) - a_i \qquad \forall i \in [\![1,n]\!], \tag{9d}$$
$$w_i \geq a_i - (t_i + W) \qquad \forall i \in [\![1,n]\!], \tag{9e}$$
$$0 \leq g_i \leq \theta \qquad \forall i \in [\![1,n]\!] : x_{i,n+1} = 1, \tag{9f}$$
$$a_i, h_i, w_i \geq 0 \qquad \forall i \in [\![1,n]\!], \tag{9g}$$
$$a_0 = h_0 = 0. \tag{9h}$$

Instead of feeding the above formulation to a solver, we can easily determine its optimal solution by inspection, if it exists. Algorithm 1 details the necessary steps and the following result proves its correctness. If, however, the subproblem is infeasible, then we need to cut off the master solution.

Proposition 4.1. *Algorithm 1 constructs an optimal solution for the reduced second-stage problem if it exists and otherwise returns a (valid) feasibility cut.*

Proof. The general approach works as follows: First, let K be the number of active teams (tours) determined by the traversal variables; i.e., $K := \sum_{i\in[\![1,n]\!]} x_{0,i}$. Second, the set of first nodes visited in all tours after leaving the depot are collected in the set $I := \{i : x_{0,i} = 1\}$. Similarly, the set of last visited nodes before going back to the depot are collected in $J := \{j : x_{j,n+1} = 1\}$.

Observe that there is a unique path constructed by the first-stage from only one node in I to a corresponding node in J, given by the k-th route. The sequence of arcs form the k-th route, a complete tour starting and ending at the depot, is denoted by $X[k]$; namely

$$X[k] \coloneqq \big((i,j) : \text{is part of the } k\text{-th route from } I \text{ to } J\big), \tag{10}$$

for all $k \in [\![1, K]\!]$. Now, a simple propagation of initial values of arrival time variables a in (9b) allows us to build a solution. For instance, consider a fixed route $k \in [\![1, K]\!]$ given by the path $X[k] = \big((0,i),(i,j),\ldots,(\ell,n+1)\big)$. As $a_0 = h_0 = s_0 = 0$, we have $a_i = \tau_{0,i}$ for the first visited node i in path $X[k]$. This initial value can be then replaced in the inequalities for the idling and waiting times at the node, such that these are satisfied; i.e., $h_i = \max\{0, (t_i - W) - a_i\}$ and $w_i = \max\{0, a_i - (t_i + W)\}$. Once this is done, the next consecutive node j in the same path is considered. In this case, the value of a_j is again set such that (9b) is satisfied. This can be done recursively for all the remaining nodes in the path and for all paths until the terminal node is reached, at which point overtime is computed using the maximum value of the arrival times at the duplicate depot $\{n+1\}$. By construction, the solution attained is not only feasible but optimal, as any increment in a_i to reduce h_i or w_i in the initial nodes $i \in I$ propagates into higher objective costs and any decrement in any variable results in an infeasible solution. □

Observe that the reduced second-stage model (9) can have multiple solutions as there is no effect of splitting waiting and idle times in a sequence of visits.

Algorithm 1 Closed-form solution for the reduced 2nd stage variables a, g, h, w for given x and t

Let $I = \{i : x_{0,i} = 1\}$, $J = \{j : x_{j,n+1} = 1\}$, and $K = |I|$.
Retrieve $X[k]$ as in (10) for all $k \in [\![1, K]\!]$.
for $i \in I$ **do**
    $a_i = \tau_{0,i}$
    $h_i = \max\{0, t_i - W - a_i\}$
    $w_i = \max\{0, a_i - t_i + W\}$
for $k \in [\![1, K]\!]$ **do**
    for $(i,j) \in X[k]$ with $i \neq 0$ **do**
        if $j = n+1$ **then**
            $g_i = \max\{0,\ a_i + s_i + \tau_{i,n+1} - L\}$
            if $g_i > \theta$ **then**
                Return feasibility cut: $\displaystyle\sum_{(s,r)\in X[k]} x_{s,r} \leq \big|X[k]\big| - 1$
        else
            $a_j = a_i + h_i + s_i + \tau_{i,j}$
            $h_j = \max\{0, t_j - W - a_j\}$
            $w_j = \max\{0, a_j - t_j + W\}$

4.5 Benders Cuts

In what follows, we present two specialised Benders cuts for determining optimality and feasibility certificates provided by the subproblems.

If any subproblem under a specific scenario is feasible, a Benders optimality cut $u(b - Ax) \leq \varphi$ is generated using the dual multipliers from the subproblem; i.e., the vector u. For each

subproblem under a particular scenario, infeasibility only occurs when the duration of any generated tour incurs an overtime. That is, the total travel, service, and idling times for a service team visiting a sequence of customers exceeds the maximum time given, plus the penalised overtime allowance. Therefore, any feasibility cut is equivalent to a subtour elimination cut with respect to the overloaded customer visiting sequence $0 \to i_1 \to \ldots \to i_\ell \to n+1$ for $\{0, i_1, \ldots, i_\ell, n+1\} \in S', \ell \in \mathbb{N}_{\geq 1}$. The following subtour elimination cut is generated to forbid the formation of such an overtime tour:

$$\sum_{\substack{i,j \in S' \\ (i,j) \in A}} x_{i,j} \leq |S'| - 1, \tag{11}$$

which can be further strengthened into the following feasibility cut, removing the two traversal arcs from depot to the first customer and from the last customer back to the depot. In this way, we are searching for an overloaded chain here:

$$\sum_{i,j \in \overline{S}} x_{i,j} \leq |\overline{S}| - 1, \qquad \overline{S} \subset S', |\overline{S}| \geq 2; \tag{12}$$

where the left-hand side of the constraint showcases an overloaded customer chain $c_i \to \ldots \to c_j$ that can occur in multiple additional extended tours. Once we identify $\overline{S} = \{c_i, \ldots, c_j\}$ as an infeasible customer set that creates an overtime tour, any longer tours containing this exact chain must also yield infeasibility. Therefore, for any overtime customer chain, constraint (12) has a broader applicability than (11) in terms of forbidding time-infeasible tours, for the latter can only forbid a single overtime tour formed by precisely one set of customers including the depot. Here, $|\overline{S}| \geq 2$ comes from the value settings that ensure that a tour must be capable of serving at least one customer. The following result proves that the strengthened version (12) is stronger and can dominate (11).

Proposition 4.2. *Given a customer chain $i_1 \to \ldots \to i_\ell$ that exceeds the overtime limit, any extended tour formed by inserting additional customers before and/or after the chain cannot restore the tour to feasibility.*

Proof. For the strengthened constraint (12) to hold, any longer tours involving additional customers before or after the customer chain cannot result in earlier arrival times as the tour gets back to the depot. Without loss of generality, we assume any customer j_* appears before the customer chain and k_* appears afterwards. We can then formulate the extended tour as $j_1 \to \ldots \to j_a \to (i_1 \to \ldots \to i_\ell) \to k_1 \to \ldots \to k_b$ containing the given overtime customer chain. Since all customer service times are strictly positive, any two chronologically and consecutively visited customers i and j should satisfy the appointment time inequality $t_i \leq t_j$.

We start by including a single customer on the extended tour. When $a = 1$ and $b = n+1$, one additional customer is visited after the depot and ahead of the first customer in the chain. Depending on the values of the team idling time h_{i_1} and customer waiting time w_{i_1} at the chain's first stop i_1, we have the following three cases: (a) When $h_{i_1} = w_{i_1} = 0$, the service at i_1 begins immediately after team arrival. The total time required to include j_1 can be written as $\tau_{d,j_1} + h_{j_1} + s_{j_1} + \tau_{j_1,i_1} > \tau_{d,j_1} + \tau_{j_1,i_1} \geq \tau_{d,i_1}$ because of $h_{j_1} \geq 0$, $s_{j_1} > 0$ and the triangular inequality. Hence, the extended tour with inserted customer j is also overtime. (b) When $h_{i_1} > 0$ and $w_{i_1} = 0$, the team arrives prior to i_1's earliest availability, causing the team to wait. In this case, it is possible for the respective team to serve at least one more customer beforehand and still encounters idling time at i_1. This can be written as $\tau_{d,j_1} + h_{j_1} + s_{j_1} + \tau_{j_1,i_1} \leq t_{i_1} - h_{i_1}$.

The service starting time however will not be any earlier than what was initially scheduled for the chain. Hence, the duration of the extended tour must be as long as the original overtime chain tour. (c) When $h_{i_1} = 0$ and $w_{i_1} > 0$, the team arrives late, thus causing the customer to wait until the start of service. The extended tour is strictly longer than the overtime chain tour by reusing the inequality under case (a). When $a = 0, b = 1$, one additional visit occurs after the chain of customers is served. Since $s_{k_1} > 0$, the extended tour must be longer than the chain tour.

Any tour with $a \geq 1$ or $b \geq 1$ can be treated as a further extension by considering the additional customers one by one using the above cases. This means that the extended tour's completion time cannot be any earlier than the original customer chain tour. In conclusion, any insertion of additional customers before or after an overtime chain tour cannot create an earlier finishing time than before and so overtime is ensured. □

Notice that our Benders model does not have complete recourse, for the subproblems in our case are not always feasible. Feasibility cuts are generated to tackle any infeasible subproblem. During the computational experimentation, it was observed that a large proportion of RMP solutions contained both feasible and infeasible tours. To enhance the MP algorithm's efficiency, we also apply (12) to prevent a group of customers that previously has formed illegal customer cycles to be allocated to the same tour at the master level, without entailing considerations regarding overtime from as in the subproblem level.

4.6 Maximum Feasible Subsystem

The Benders feasibility cuts are generally considered unpreferable, for they bring no improvement to the lower bound in the RMP. Slow convergence of the Benders' lower bound is most often due to the creation of numerous infeasible solutions that yield many feasibility cuts before a feasible solution occurs and produces an optimality cut. In order to generate optimality cuts earlier on to accelerate the convergence of the Benders' lower bound, we adapt the idea of a *maximum feasible subsystem* (MFS) from [SI10] so that each time a feasibility cut is produced, we generate alongside an additional MFS cut to return to the current RMP.

The MFS cut is formed by relaxing a minimum number of SP constraints to restore the feasibility of the subproblem. Since any infeasible subproblem in our case can only be caused by overtime, we identify the only unsatisfied constraint in the subproblem as (2g). By relaxing it, we can derive an MFS containing (2b)–(2f), (2h), and the following constraint to restore that particular scenario to feasibility:

$$g_i \leq \theta + M' \qquad \forall i \in [\![1, n]\!]; \tag{13}$$

where M' is a sufficiently large number to make sure that any overtime tour can now fit into the service team time capacity. We can then generate an MFS cut in the same way as with an optimality cut via computing the dual multipliers of the relaxed primal subproblem. The dual multipliers associated with the relaxed constraint (13) can be fixed to zero due to complementary slackness. As a result, an infeasible scenario can generate one feasibility cut and one MFS cut, which shares similar structure with an optimality cut.

Even though Rahmaniani et al. [Rah+17] suggests the strategy can be non-competitive due to the additional computational cost for finding the MFS compensating the gain in fewer cut iterations. This weakness however does not apply to our implementation because we know precisely which constraint will be infeasible. Therefore, there is no need to iterate through all the subsystems to identify the maximal one. Besides, our closed-form subproblem solution ensures that our computational time remains linear with respect to the node size.

4.7 Minimum Infeasible Subsystem

Another issue associated with slow convergence of the Benders bound is the poor quality of cuts generated in each iteration, especially at the beginning of the Benders algorithm. For the combinatorial Benders feasibility cut (12), the effect can be relatively weak, especially if the size of the set S' is large with a large excess in overtime. Therefore, we strengthen the feasibility cuts based on the *minimum infeasible subsystem* (MIS) from [CF06], which is a subset of customers that cannot be served together by a single service team without overtime but the removal of any node results in a feasible single tour.

Enumerating through every existing MIS for a particular customer set would be highly computationally inefficient. To derive stronger MIS feasibility cuts, we start with the tours formulated in the MP. We use a local-search-based heuristic for each tour within the solution to find the set of minimum infeasible subsets that are slightly overtime, yielding stronger feasibility cuts. We adapt a local search procedure based on the current status of the tour, to iteratively remove or insert customer nodes to create a MIS and return relevant cuts, see Algorithm 2.

Algorithm 2 Finding MIS

```
Let tour be infeasible
while tour is infeasible do
    MIS_tour ← tour
    candidate ← ∅
    for node ∈ tour do
        aux_tour ← tour.Remove(node)
        if aux_tour is infeasible then
            gain ← cost(tour) − cost(aux_tour)
            candidate ← candidate.Append(node, gain)
    node ← node with maximum gain from candidate
    tour ← tour.Remove(node)
Return MIS_tour
```

5 Heuristic Algorithm

Realising the difficulty of tackling the problem as a whole, 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. 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.

5.1 Description of the Method

Our two-stage heuristic resembles a typical home service rundown: previous-day initial plannings (Section 5.3), service day tour refinements (Section 5.4), and post-service performance evaluation (Section 5.5). The computational efficiency positions our heuristic as an effective alternative for mitigating the impact brought by customer cancellation (Section 5.2), leveraging available information to generate high-quality decisions during the decision-making process. The heuristic showcases an "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 4 displays an example for the two-stage heuristic outputs.

Initial Planning Stage represents the real-time circumstances when the decision-makers make pre-arrangements to work diligently to ensure a smooth rundown on the actual service day with incomplete information. The available information at this stage includes the customers' geographical locations, the expected probability of cancellations, and estimated travel and service times (from known distributions or historical data). Before the heuristic begins, we aim to determine a fleet of available teams m on a strategic level based on methods such as the root node approach or estimations. At the initial stage of the heuristic, we aim to determine a workable fleet size $\widehat{m}$ to use and construct a set of *a priori* routes for the service teams. We also aim to notify the customers of an initial appointment slot for the service. The decision-makers are free to select a specific fleet size within the feasibility interval according to their preference or risk-averseness, and the fleet size will then be optimised by the heuristic (see Figure 4 for an example).

Tour Refinement Stage resembles the actual service day. With all in-time customer cancellations obtained before a given cut-off time, the second stage provides a refinement of the solution from the first stage. After cancelled customer nodes are removed from the tours, the cluster and the visiting sequences are re-optimised for more balanced workload among the service teams. All non-cancelled customers will be notified of an updated and narrower appointment time window at the beginning of the service day. Specifically, we aim to design the routes such that all the second-stage updated appointment time windows lie within their associated initial time windows derived in the previous stage (see Figure 4 for an example). Additional scenarios are generated to evaluate the robustness of the second-stage solutions.

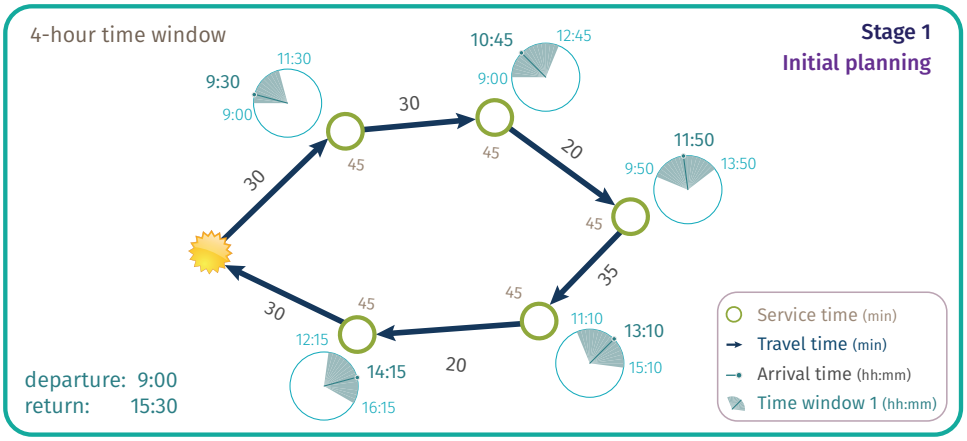
Post-service Evaluation Stage possesses complete information with specific travel and service durations revealed after the actual service. This is a post-service reflection period for the service providers to evaluate the service teams' performance.

5.2 Customer Cancellation Uncertainties

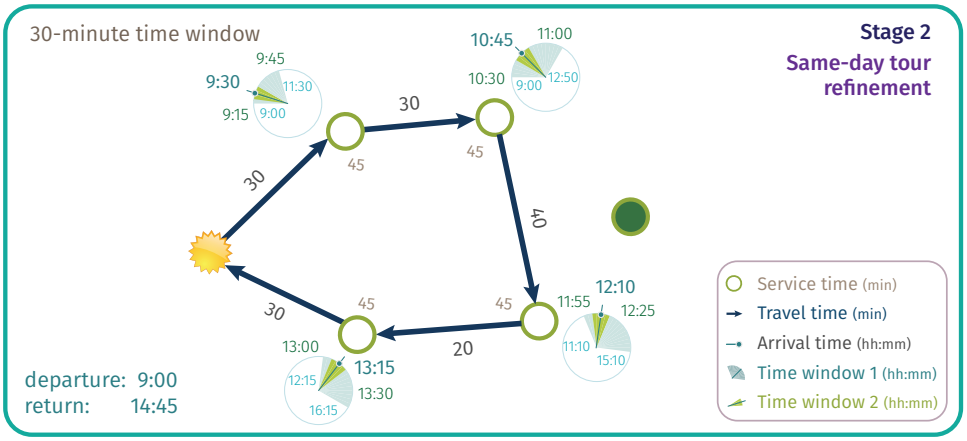
For the heuristic, while the in-time customer cancellations can be directly addressed during the tour refinement stage, the heuristic aims to deliver sound and up-to-date decisions to ultimately mitigate the impact of any other type of customer cancellations occurring after the cut-off time, as well as the uncertainties related to travel and service times. We come up with three customer cancellation scenarios on the service day and their associated policies as below. A cancellation policy analysis is given in Section 6.5.

1. *In-Time Cancellation*: Any cancellation that occurs before a fixed cut-off time or the departure of the assigned team from the depot, whichever is earlier. Any such customer is skipped by the team, which will travel from its previous stop directly to the next scheduled customer, whose appointment time is adjusted;

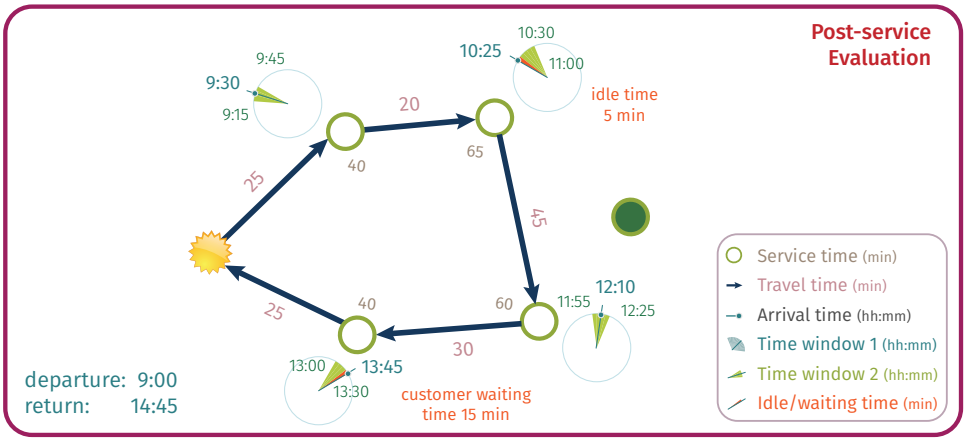
2. *Last-Minute Cancellation*: Any cancellation occurs when the team is at any service stage but before its departure for the cancelled customer from the previous visit. The additional travel to the customer cannot be avoided due to short notice, but the team spends zero time (i.e., service duration is 0) at such location;



- For a specific team, plan initial visiting sequence and arrival times for all assigned customers based on estimated travel and service times
- Compute initial appointment time windows using arrival times



Remove cancelled customers and update time windows (green) based on estimated travel and service times



Receive travel and service times realisations, compute idle/wait/overtime penalties, and evaluate service team's perfomance and solution robustness

Figure 4: Heuristic framework: initial planning, tour refinement, and post-service performance evaluation stages

3. *No-Presence*: Considered as a "no-show" or "irresponsible customer" who does not notify about the cancellation. The team arrives at the customer assuming regular service and waits for a short period (less than service time) before registering the cancellation and thus chooses to leave for the next stop. Additional idling time occurs.

5.3 Initial Planning Stage

The initial planning stage consists of four consecutive steps: customer districts construction, district-based routes construction, workload balance improvement, and initial appointment scheduling. The graphs in the left-hand side column of Figure 5 demonstrate the first-stage decision planning process. For the decision-makers, the initial solution provides the number of service teams to use, the sequence of customers for each service team, and an initial 2-hour appointment slot assigned to all existing customers. The solutions are generated based on estimated traversal and service times (see the next section for details) and only considers in-time cancellations.

5.3.1 Estimate service and travel times with probabilistic customer presence

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 servable 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 can be inaccurate without knowing the actual cancellation list. Notwithstanding, we can estimate the travel and service times for probabilistic customers without explicit routing as proposed by Bard and Jarrah [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 distributions associated to travel and service times. 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 following formula given in [BJ09]:

$$e_i = \sum_{j=1}^{b_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{v_{i,j}} \quad = \sum_{j=1}^{b_i} \frac{(1-q_j)(b_i - R_{i,j}+1)}{\sum_{\ell=1}^{b_i}(1-q_\ell)(b_i - R_{i,\ell}+1)} \cdot \frac{d_{i,j}}{v_{i,j}} \tag{14}$$

where q_j is customer j's probabilistic cancellation rate, b_i is the fixed number of closest customers to customer i, $R_{i,j}$ is the rank of the j-th closest customer with respect to i, with $j \in [\![1, b_i]\!]$. $p_{i,j}^{(*)}$ 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, AM_i, for customer i can be estimated by the expected service time $\widehat{s}_i$ plus the estimated travel time from i to the district centre j using (14). Here we use the expected travel velocity $\widehat{v}$. 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 that for a specific customer i:

$$\mathsf{AM}_i := \widehat{s}_i + e_i = \widehat{s}_i + \sum_{i=1}^{b_i} p_{i,j}^{(*)} \cdot \frac{d_{i,j}}{\widehat{v}_{i,j}} \tag{15}$$

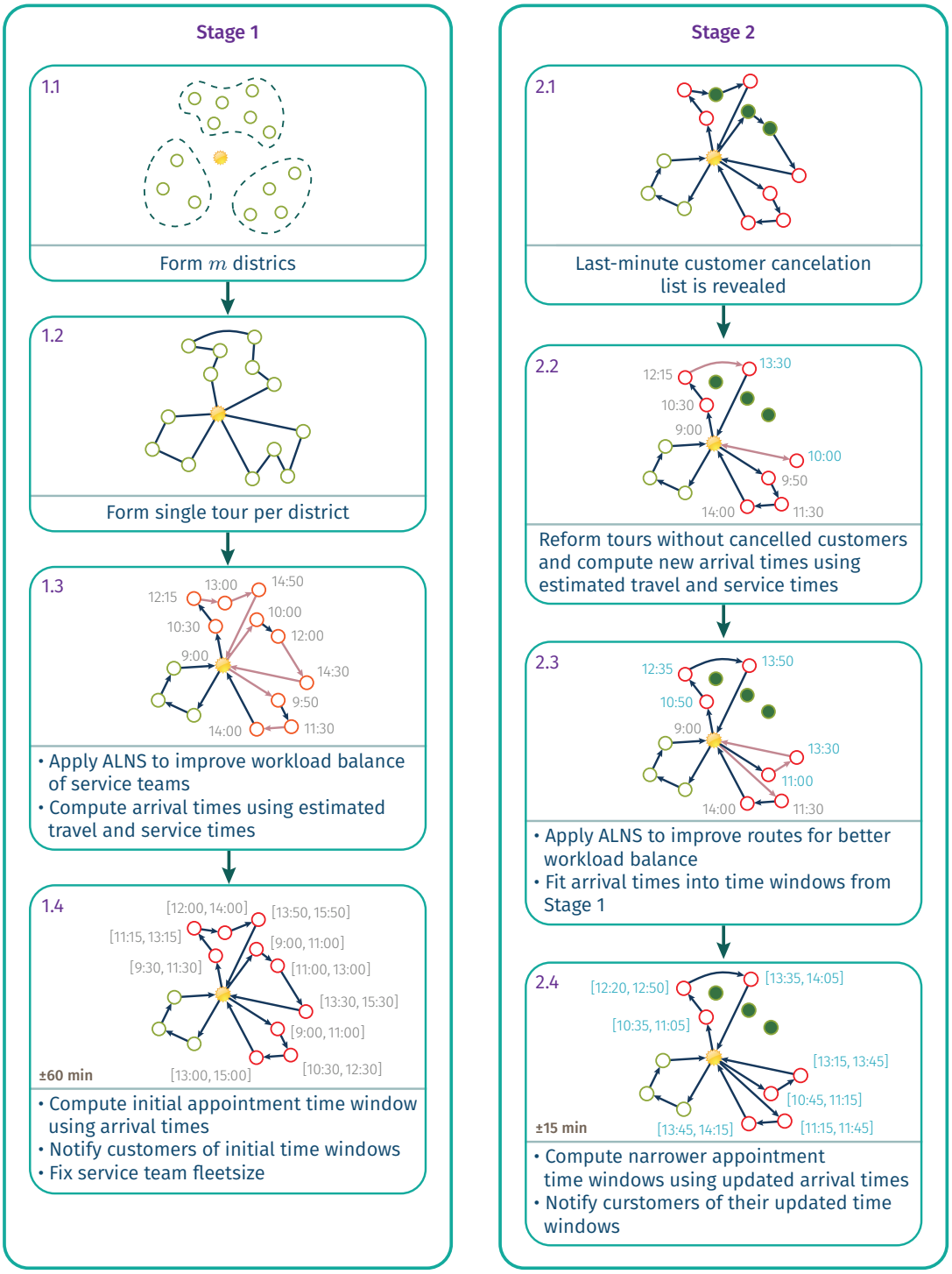


Figure 5: Example of the Initial Planning Stage and Tour Refinement Stage

Here, we use the average number of nodes within a district. However, we observe that b_i is sensitive to the cancellation rate, for a high cancellation indicates that the selection of the next customer can exhibit a significantly random pattern. In contrast, a low cancellation makes the next node to be visited almost certain and establishes an inverse relationship with the distance.

5.3.2 Customer districting (Stage 1.1)

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. Mathematically, we first aggregate customers into m compact and balanced districts that are each manageable by an individual service team, then we develop explicit routes within each district. A feasible fleet size m can be pre-determined using the root-node solution method described in Section 4.3.2.

We adapt the districting formulation proposed by Hess et al. [Hes+65] and solve the resulting MIP model to optimality to receive our initial customer-team assignment decisions. Let $[\![1, n]\!]$, be the set of customers and $\{0\}$ be the depot as before. Let $\mathsf{AM}_i \in \mathbb{R}^+$ 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 then as $\mu := \frac{1}{m}\sum_{i\in[\![1,n]\!]} \mathsf{AM}_i$. We denote $\mathsf{AM}_{\min} \leq 100$ and $\mathsf{AM}_{\max} \geq 100$ as the minimum and maximum percentage of activity measures in a district, respectively. 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_{jj} takes the value of one if customer j is selected to be the district centre. The districting MIP model can be defined as below:

$$\min \sum_{j\in[\![1,n]\!]} \sum_{i\in[\![1,n]\!]} \mathsf{AM}_i \, d_{i,j}^2 y_{i,j} \tag{16a}$$

subject to

$$\sum_{j\in[\![1,n]\!]} y_{i,j} = 1 \qquad \forall i \in [\![1, n]\!], \tag{16b}$$

$$\sum_{j\in[\![1,n]\!]} y_{j,j} = m \tag{16c}$$

$$y_{i,j} \leq y_{j,j} \qquad \forall j \in [\![1, n]\!], \tag{16d}$$

$$\sum_{i\in[\![1,n]\!]} \mathsf{AM}_i \, y_{i,j} \geq \frac{\mathsf{AM}_{\min}}{100} \mu \cdot y_{j,j} \qquad \forall j \in [\![1, n]\!], \tag{16e}$$

$$\sum_{i\in[\![1,n]\!]} \mathsf{AM}_i \, y_{i,j} + 2d_{0,j} \leq L \qquad \forall j \in [\![1, n]\!], \tag{16f}$$

$$y_{i,j} \in \{0, 1\} \qquad \forall i, j \in [\![1, n]\!]. \tag{16g}$$

Constraints (16b) require every customer to be assigned to a district. Constraint (16c) requires exactly m districts to be formed. Constraints (16d) state that each formed district must have a center. Constraints (16e) define the minimal workload of any district. Constraints (16f) stress that the workload (activity measure) within each district, together with the pendulum tour to and from the depot, cannot exceed the total time allowance.

5.3.3 Routing within individual districts (Stage 1.2)

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 *travelling salesman probelm* for

m times, where m is the number of service teams. We adapt the TSP model formulated by Dantzig et al. [DFJ09] to receive our initial routing decisions:

$$\min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} d_{i,j} x_{i,j} \tag{17a}$$

subject to

$$\sum_{\substack{i \in [\![1,n]\!] \\ i \neq j}} x_{i,j} = 1 \qquad j \in C, \tag{17b}$$

$$\sum_{\substack{j \in [\![1,n]\!] \\ i \neq j}} x_{i,j} = 1 \qquad i \in C, \tag{17c}$$

$$\sum_{\substack{i,j \in Q \\ i \neq j}} x_{i,j} \leq |Q| - 1 \qquad Q \subsetneq C, |Q| \geq 2, \tag{17d}$$

where $x_{i,j}$ is a binary variable indicating that arc (i, j) is traversed. Constraints (17b) and (17c) guarantee the flow conservation that each node should has exactly one vehicle arriving and leaving. Constraints (17d) eliminate sub-cycles containing only a subset of nodes. A comprehensive review on the TSP heuristics methodologies and implementations can be found in [Reg+11]. Taking into account the size of our problem, an exact solution can be obtained using existing solvers.

5.3.4 Improving the initial routes (Stage 1.3)

So far we have constructed the initial set of routes with the cluster-first-routing-second strategy. To improve upon these routes, we employ the *adaptive large neighbourhood search* (ALNS) metaheuristic. 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". The method looks for a better solution by destructing a part of the solution and reconstructing the damaged part in a different way. The implementation of this metaheuristic and its variations have yielded promising results on some complicated VRP variants with large instances [RP06; HCC12]. Our pseudocode for the ALNS is presented in Algorithm 3.

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. The first three destroy operators are at the customer node level, while the latter three are at the routing level. We set by default the choice of $q = 5$ from our experimental results.

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:* a group of q customers with the highest removal gain are selected and removed from their existing routes. The removal gain refers to the difference of cost of having or not a customer inside a given allocated tour.

Algorithm 3 Basic steps of ALNS

```
s ← Initial_Solution, Initial_Score(w*) and s^best = s
while stopping criteria not met do
    N⁻ ← Choose(All_Destroy_Operators, w*_d)
    N⁺ ← Choose(All_Repair_Operators, w*_r)
    s' ← Destroy_Repair_Apply(s, N⁻, N⁺)
    if s' < Quality_Threshold then
        s' ← Local_Search(s')
    obj(s') = sum cost (team, travel, overtime)
    if s' satisfies an acceptance criterion then
        s ← s'
        if s' < s^best then
            s^best ← s'
    update Roulette_Wheel operators performance scores
```

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 removes a single tour and moves all the allocated customers from this single tour to the customer pool.

5. *Longest Tour Break into Half:* breaks the longest tour found into two smaller tours of the same length (or with one containing one more customer). Link the starting and ending customers of the smaller tours to the depot.

6. *Overcapacitated Tour Break into Half:* breaks all the infeasible tours (time capacity exceeded) in the middle and form two smaller tours. Link the start and end of the smaller tours with the depot.

After the destruction process, all customers inside the customer pool will be re-inserted by a 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 in the interval [0.8, 1.2].

3. *Greedy Insertion Forbidden:* Once again, the same mechanism as *Greedy Insertion*, only that a customer node cannot be re-inserted to the same position it has just been removed from.

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% from initial experiments). A description of the local search methods is given in Figure 6.

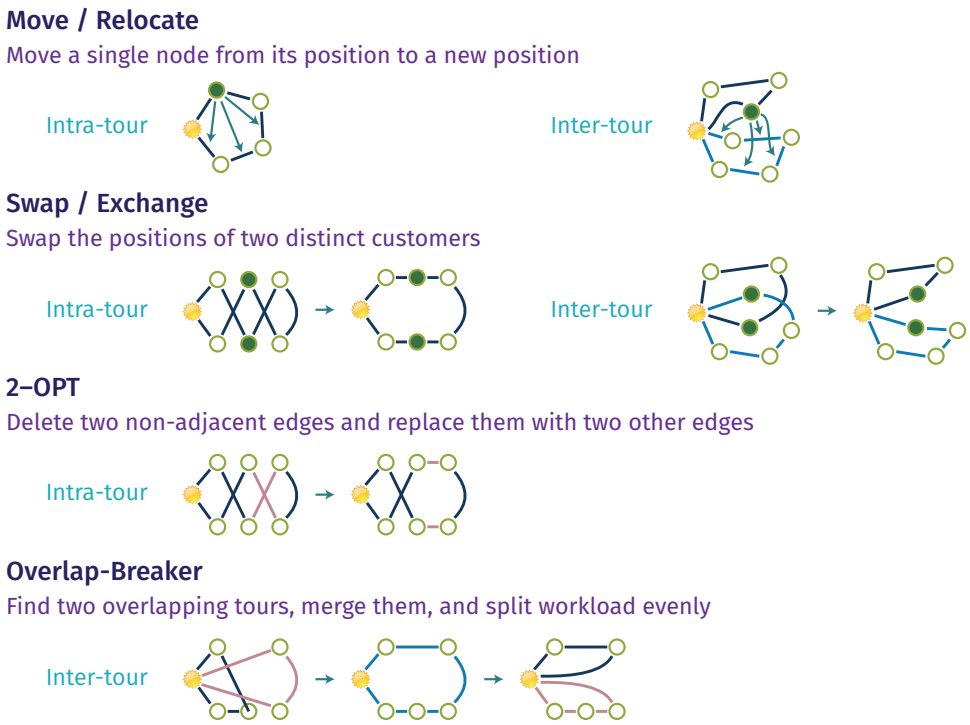


Figure 6: Local Search Operators: a graphical description of the *move*, *swap*, and *two-opt* methods

Since the destruction and reconstruction process (with local search) allow us to modify the number of existing tours, it is therefore possible to re-optimise the fleet size during the ALNS search. Consequently, ALNS allows our first-stage heuristic to be less affected by a fixed service team number m.

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 with index i is selected with probability $\mathsf{ow}_{i,j} / \sum_{k=1}^{K} \mathsf{ow}_{k,j}$, where K is the group of same-category operators and j is the segment index. A segment is a consecutive number of iterations during the solution process. During a segment, the i-th operator is associated with a weight $\mathsf{ow}_{i,j}$ to reflect its capability to bring improvement to the incumbent in the current segment j. The weight of each operator is periodically updated using a weighted balance of the previous N iterations and the overall historical performance throughout the ALNS process. We compute the weight of the i-th operator in segment $j+1$ via:

$$\mathsf{ow}_{i,j+1} = \begin{cases} \mathsf{ow}_{i,j}(1-r) + \frac{\pi_i}{Q_i} r & \text{if } \pi_i > 0, \\ \mathsf{ow}_{i,j} & \text{if } \pi_i = 0, \end{cases} \tag{18}$$

where $\mathsf{ow}_{i,j}$ is the weight of the i-th operator in the previous segment j. π_i is the score that the operator earned in segment j for contributing to the improvement of incumbent quality, and Q_i is the number of times the i-th operator has been employed. Thus $\frac{\pi_i}{Q_i}$ is the average score the i-th operator earns each time it is 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 our computational experimentation.

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 $\exp\{[f(s')-f(s)]/\mathsf{Tem}\}$ where s is the current solution and Tem 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.

5.3.5 First-stage appointment scheduling (Stage 1.4)

The last step of the first-stage heuristic is to notify all customers of their initial appointment time windows. Based on the set of ALNS-refined routes, we derive each individual appointment time for a customer from the associated team's arrival time at the customer using the expected travel and service times. To cope with potential customer cancellations and unpunctuality, we expand each individual appointment time into an appointment time window with fixed length T_1 and quote 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 this customer will be [9:30, 13:30] am.

5.3.6 Further Improvements

To further improve on the real-life practicality of our routes and schedules derived after the districting-first-routing-second construction heuristic from Sections 5.3.2 and 5.3.3 and the ALNS improvement heuristic from Section 5.3.4, we consider the following improvements for our first-stage solution: (i) team workload balance, (ii) multiple tours overlapping minimisation, and (iii) twisted tour elimination.

Each service team's assigned workload is bounded by (16e) and (16f), 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 $P \cdot \max\left\{\mu^{-1}|\sum_{i\in k}\mathsf{AM}_i - \mu| - \alpha, 0\right\}$ 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. The randomness then challenges the algorithm to form disjoint, compact, and contiguous driver zones within a reasonable computing time. However, the application of *overlap-breaker* and *2-opt* (cf. Figure 6) can remove the majority of overlaps and eliminate twisted tours that self-intersect.

5.4 Tour Refinements Stage

We make initial routing decisions during the first-stage heuristic without knowing which customers will remain in the system. At the beginning of the second stage an updated list of cancelled customers $\mathbb{I}$ becomes known. So we re-optimise the initial tours to fit the up-to-date

customer information. The graphs in the right-hand side column of Figure 5 shows the decision process for our second-stage tour refinement: we remove cancelled customers from the first-stage tours, compute the estimated arrival times for all non-cancelled customers, improve service teams workload balance, and notify all non-cancelled customers of a narrower appointment time window.

5.4.1 Second-stage Routing Improvements (Stage 2.3)

At the beginning of the tour refinement stage, the list of newly cancelled customers is revealed and removed from the first-stage tours. From the right-hand side column of the diagram in Figure 5, the second stages 2.1 and 2.2 are relatively straightforward. For Stage 2.3, we re-apply the ALNS improvement heuristic introduced in Section 5.3.4 to refine the routes and customer schedules. The differences in implementing ALNS at this stage are twofold.

First of all, in contrast to the expected travel and service times from known distributions used inside the first stage computation, we use a scenario-generated set of travel and service times at the second stage. Secondly, the objective function in the previous stage consists of total costs from team hiring, travelling, and working overtime. In contrast, our second-stage objective function consists of the averaged total costs from travelling, team hiring, and overtime plus penalties for workload imbalances and scheduling the new arrival times outside the first-stage appointment time windows. The last term encourages the algorithm to create a nested structure for the updated appointment times to lie within the previously determined time windows, which is essential to service quality because abrupt modifications to service starting times are highly unfavourable with customers.

5.4.2 Second-stage Appointment Scheduling (Stage 2.4)

The service teams' arrival times to customers and the 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 yet again a time to every non-cancelled customer, albeit a narrower one. Having the second-stage time window nested within the first-stage time window is essential for successful service and thus customer satisfaction. Specifically, we assume we have 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. We add to the ALNS objective term $P^{'} \times \max\{T_1^{\text{start}} - a_i, a_i - T_1^{\text{end}}, 0\}$, which penalises any arrival time not nested within the first-stage time window. Similar to the first-stage appointment scheduling, we create a narrower second-stage time window of length $T_2 = 30$ min. Moreover, the time windows are not necessarily centred at their arrival times. This is determined by a linear adjustment $[a_i - T_i^{\text{start}}] \cdot c_{\text{idle}} = [T_i^{\text{end}} - a_i] \cdot c_{\text{wait}}$ that forces the center forward in time to cope with more expensive waiting costs, or backward with more expensive idling costs.

5.5 Out-of-sample Performance Evaluation

We notice that the issue of data over-fitting might occur in our two-stage heuristic framework, since we only rely on in-sample objective values computed using a discretised set of scenarios n_e clustered from random samples. To avoid the solution being over-fitted to the in-sample population, we use a new set of benchmark scenarios to evaluate the additional out-of-sample performance of our second-stage solutions. This gives a fairer indication of how good the proposed level of service (in terms of cost and penalties) obtained from the heuristic is on an

unobserved set of data. If the out-of-sample level of service deviates significantly from the in-sample level of service, then this indicates that the scenarios generated for the tour refinement stage are not a suitable representation of the uncertainties and, thus, the planning should be repeated with a newly generated set of scenarios.

6 Experiments

6.1 Experimental Setting

Since our problem setting is motivated by an application in home healthcare, we consider the parameter settings that were given in the AIMMS-MOPTA competition guidelines [SC21]. 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 $c_{\text{over}} = 5$, and hourly customer waiting cost $c_{\text{wait}} = 4$. We also define the standard daily workload $L = 8$ hours for an individual team, the first-stage time window length $T_1 = 2$ hours, and 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. This distribution has been applied in many existing studies for its recognition of the skewed distributions in modelling travel times [Gut+18]. 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 $\widehat{s} = \mu_s = 45$ min with its standard deviation set to $\mu_s/2$, the expected travel speed $\widehat{v} = 1$ km/min (equivalently, expected travel time $\tau_{i,j} = 1$ min/km). The traversing time on arc (i, j) can be computed by the travel time per unit distance multiplying the Euclidean distance. Moreover, we assume all customers share the cancellation probability defined at a fixed rate 5%.

We implemented our Benders algorithm, CPLEX exact methods algorithms, and our two-stage heuristics framework in Python. For a unified measurement, we use CPLEX 20.1.0 as the optimisation solver for both exact methods and heuristics. The computations are performed on a machine with Intel Gold 6234 CPU and 512GB RAM installed. We used a single-core setting to run a total of 10 test instances.

A Sampling-Based Objective Function Since the travel and service durations as well as the customer cancellation list are random, 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 [SS09]:

$$f^*(x, t) = \frac{1}{n_e} \sum_{i \in [\![1, n_e]\!]} f\big(x, t, S_i(\tau, s)\big) \tag{19}$$

where $f^*(x, t)$ is the expected total cost 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 ith scenario with stochastic travel and service times realisation. $f\big(x, t, S_i(\tau, s)\big)$ represents the total costs of the ith scenario applied to x, and finally n_e is the total number of scenarios. The sampling-based objective function introduced above in (19) is used to compute the first-stage objective in (1a) for the experiments.

Monte-Carlo Simulation and Scenario Generation We introduce a scenario-generating procedure to include a more diverse set of scenarios, and meanwhile, we wish to maintain a reasonable computational complexity by limiting the number of scenarios.

First, we assume a number of n_e scenarios and apply Monte-Carlo simulation to randomly generate n_s samples, each with a different pair of travel and service times realisations. If $n_s = n_e$, then each sample is an individual scenario. However, a large number of samples is usually required to leverage a probability distribution and project outcomes more accurately. Therefore, we generate a sufficiently large set of samples ($n_s \gg 30$) because of the *law of large numbers*, which states that the average experimental results converge to the expected value with a large number of trials. Mathematically, we can treat each generated sample as a vector consisting of the travel time realisation for every traversed arc and the service time realisation for every non-cancelled customer node. For example, for a second-stage solution with two teams travelling 8 arcs to serve 6 customers, the sample will be a vector of 14 values (8 travel times + 6 service times). Each value has been individually and independently drawn from the associated known distributions.

We then partition these n_s samples into n_e clusters using a k-means clustering algorithm. The sample at the center of each cluster is then taken as the representative scenario. The probability q^ω of each scenario ω is computed as 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 Comparison of Exact Methods

We implemented the CPLEX stochastic full model algorithm (**MIP-full**), CPLEX built-in Benders algorithm (**Benders-C**), and our customised Benders framework (**Benders**). MIP-full tackles the MIP model as a whole until an optimal solution is found or when the computational time limit is reached. Benders-C uses CPLEX's built-in Benders algorithm with the same decomposition framework but works as a black box. Our customised Benders algorithm begins with a RMP that is iteratively checked by subproblems and has its feasibility region reshaped by optimality and feasibility Benders cuts until the MP converges.

We compare the performance of the three algorithms and report the results in Table 1. The following Table 2 demonstrates the in-sample and out-of-sample analysis for the algorithm performance. The maximal computational time for all three algorithms is fixed to be 1 hour. Due to the variations in service time, travel time, and cancellations between different generated scenarios, we run each customer $|C|$ and scenario $|S|$ combination for 10 times and return the averaged performance data. We apply the ALNS metaheuristic warm-start to produce an initial feasible solution from which all the algorithms begin. The generated Benders optimality and feasibility cuts and the set of valid inequalities in our algorithm are added to the MP via the `LazyConstraintCallback`. We test the algorithms using customer numbers from $[10, 20, 30, 40]$. For the stochastic instances, we generate sets of $[20, 40, 60, 80, 100]$ scenarios for travel and service times from 200 random samples using the proposed scenario generation procedure. For each customer-scenario combination, we report the average computational time in seconds, `UB` as the best integer solution found in the branch-and-bound process, `LB` as the best lower bound found, and `Gap` as the percentage difference between `UB` and `LB`, computed as $\texttt{Gap} = 100 \times {}^{(\texttt{UB}-\texttt{LB})}/_{\texttt{UB}}\,\%$. Moreover, we also report the number of Branch-and-Bound nodes, as well as the min, max, mean and standard deviation for the in-sample and out-of-sample analysis.

In Table 1, our Benders algorithm exhibits superior performance, particularly evident in smaller instances comprising up to 30 customers. It achieves the smallest UB-LB gap across various customer and scenario sizes compared to the two alternative methods. However, as the customer size increases to 40 and coupled with more than 20 scenarios, Benders-C begins to outperform its counterparts, showcasing the most optimal gap with the most searched nodes

amongst the three algorithms. In general, it is observed that as the size of customers and scenarios increases, the Benders and Benders-C algorithms exhibit a more significant outperformance compared to the MIP-full algorithm in terms of gap.

We name instances with a certain number of customers as a critical point, where the fleet size will increase by one with any additional customers. Since the service team hiring costs dominate the travel costs and other penalty costs, increasing the fleet size by one leads to a significant difference in the total costs. During the solving process, an update in fleet size can also cause a sudden jump in `LB` when specific scenarios contain large service and travel times that make the master-level routes infeasible. The occurrence of a critical point is the reason behind a sudden decrease in the gap from $|S| = 20$ to 60 for the 20-customer instances. This is because more scenarios can appear that require the creation of another service tour to accommodate unusually high travel and service times. A larger scenario size means that more extreme scenarios like this can be included. The warm-start initial solution is more likely to reflect this by building an initial set of tours with a larger fleet size, and hence influence the evolving process of the `UB` and `LB`.

In Table 2, the observed trend in the out-of-sample compared to the in-sample results among the three algorithms remains consistent. Despite the the slightly larger standard deviation indicating greater variation in performance for our Benders, both the in-sample and out-of-sample average objective values are comparable to the alternative methods. In Table 3, we present the average Value of the Stochastic Solution (VSS) and the average Expected Value of Perfect Information (EVPI) for 10 customers across various scenarios from $[20, 40, 60, 80, 100]$.

Table 1: Comparison of CPLEX stochastic whole model, CPLEX built-in Benders and our implemented Benders algorithm (Nodes are per 1,000)

		Benders-C					**Benders**					**MIP-full**				
$\|C\|$	$\|S\|$	**Time**	Gap	LB	UB	Nodes	**Time**	Gap	LB	UB	Nodes	**Time**	Gap	LB	UB	Nodes
10	**20**	480.09	6.81%	189.50	203.32	200.41	370.36	3.59%	195.89	203.15	270.99	480.41	6.84%	189.59	203.49	51.66
10	**40**	0.34	0.5%	202.89	203.62	0.00	10.34	0.5%	202.70	203.38	0.00	1.66	0.5%	202.94	203.78	0.01
10	**60**	0.47	0.5%	202.94	203.70	0.00	16.18	0.5%	202.78	203.36	0.00	1.77	0.5%	202.99	203.78	0.00
10	**80**	1029.0	14.3%	174.55	203.63	253.47	482.89	0.5%	202.49	203.20	1079.91	1032.94	14.42%	174.35	203.66	17.05
10	**100**	0.5	0.5%	203.02	203.85	0.00	61.33	0.5%	202.76	203.41	0.00	16.36	0.5%	203.09	203.94	0.12
20	**20**	3323.12	30.52%	211.71	304.71	2975.41	3325.03	30.47%	211.66	304.38	3416.06	3323.44	30.57%	211.57	304.73	148.52
20	**40**	3600+	33.12%	204.13	305.24	1308.37	3600+	32.97%	204.17	304.61	6179.56	3600+	33.21%	203.97	305.37	61.03
20	**60**	3600+	33.22%	204.03	305.53	522.86	3600+	32.95%	204.12	304.43	2019.40	3600+	33.29%	203.90	305.63	18.36
20	**80**	3168.4	21.02%	241.71	306.02	384.90	2285.54	20.74%	241.65	304.84	3048.68	3600+	21.09%	241.59	306.16	8.11
20	**100**	3600+	36.64%	203.81	325.83	262.35	3600+	33.01%	203.93	304.41	2842.96	3600+	33.45%	203.69	306.06	5.38
30	**20**	3600+	24.97%	304.80	406.22	2255.22	3600+	24.86%	304.76	405.59	2102.88	3600+	25.07%	304.67	406.60	41.66
30	**40**	3600+	25.08%	304.72	406.72	1305.02	3600+	24.88%	304.70	405.63	3315.58	3600+	25.3%	304.56	407.70	7.92
30	**60**	3600+	29.45%	304.62	436.03	804.07	3600+	27.1%	304.58	420.30	1346.72	3600+	29.68%	304.45	437.10	2.72
30	**80**	3600+	34.13%	304.78	468.11	406.14	3600+	30.91%	304.75	446.34	1262.41	3600+	37.12%	304.64	488.18	1.14
30	**100**	3600+	33.68%	304.57	464.69	528.24	3600+	33.47%	304.54	463.31	798.71	3600+	38.02%	304.42	494.03	0.84
40	**20**	3600+	20.18%	405.58	508.11	1440.27	3600+	20.02%	405.47	506.99	308.75	3600+	20.51%	405.43	510.03	11.18
40	**40**	3600+	23.58%	405.47	533.78	918.17	3600+	25.02%	405.41	544.92	674.39	3600+	28.72%	405.33	572.79	1.47
40	**60**	3600+	29.81%	405.34	581.29	583.24	3600+	31.49%	405.25	593.90	82.55	3600+	31.86%	405.20	596.80	0.74
40	**80**	3600+	33.6%	405.42	610.58	517.36	3600+	34.39%	405.35	619.22	236.27	3600+	36.5%	405.23	644.02	0.46
40	**100**	3600+	33.49%	405.34	609.45	580.26	3600+	34.95%	405.23	624.99	109.13	3600+	38.0%	405.16	661.33	0.42

Table 2: In-Sample and Out-of-Sample performance Comparison of CPLEX stochastic whole model, CPLEX built-in Benders and our implemented Benders algorithm

		Benders-C				**Benders**				**MIP-full**			
$\|C\|$	$\|S\|$	IS-Mean	IS-Std.	OS-Mean	OS-Std.	IS-Mean	IS-Std.	OS-Mean	OS-Std.	IS-Mean	IS-Std.	OS-Mean	OS-Std.
10	**20**	203.36	0.56	203.59	1.05	203.36	0.56	203.59	1.05	203.60	0.94	203.76	1.33
10	**40**	203.69	1.20	203.89	2.18	203.88	1.52	204.11	2.54	203.87	1.81	203.98	2.18
10	**60**	203.89	2.10	203.98	2.51	204.03	2.40	204.17	2.67	204.04	2.48	203.98	2.18
10	**80**	203.89	2.31	204.24	3.36	203.98	2.24	204.42	3.73	203.99	2.35	203.95	2.25
10	**100**	204.02	2.12	204.31	2.99	204.02	2.12	204.31	2.99	204.13	2.52	204.40	3.31
20	**20**	304.75	0.61	305.02	1.37	306.65	2.53	306.82	2.72	304.77	0.63	304.96	1.12
20	**40**	305.37	1.64	306.33	4.59	306.23	3.78	306.62	5.89	305.50	2.11	305.99	3.70
20	**60**	306.07	4.58	306.78	5.31	306.52	5.60	306.91	5.60	306.11	3.30	307.00	5.07
20	**80**	306.77	4.74	306.74	5.05	307.39	6.16	307.01	5.82	307.09	5.72	306.92	5.54
20	**100**	326.32	3.77	326.67	4.83	307.23	6.33	306.83	6.10	306.64	5.29	306.66	4.92
30	**20**	406.29	0.78	406.87	2.49	408.57	2.68	409.14	3.79	406.67	1.19	406.96	2.18
30	**40**	407.07	2.56	408.15	5.23	408.26	4.21	408.81	4.87	408.12	3.71	409.01	5.78
30	**60**	436.97	5.33	437.64	6.82	423.47	6.28	423.87	7.28	437.81	4.98	438.40	6.65
30	**80**	469.29	5.89	469.52	6.20	449.88	7.02	449.91	8.71	489.35	6.51	489.73	7.20
30	**100**	465.50	4.92	466.51	6.97	466.26	5.99	467.29	8.35	494.84	5.32	495.38	6.35
40	**20**	508.18	1.29	508.65	2.06	512.36	2.82	512.42	2.90	510.19	2.23	510.44	2.59
40	**40**	534.49	4.20	534.96	4.48	548.50	5.28	548.76	6.00	573.38	4.43	573.64	6.03
40	**60**	582.24	5.26	583.35	6.91	596.07	4.73	597.04	6.31	597.84	6.31	598.01	6.66
40	**80**	611.57	5.05	612.79	7.22	621.80	5.42	622.92	7.59	645.37	6.74	646.13	8.54
40	**100**	610.46	5.00	611.68	7.96	627.80	5.89	628.74	8.09	662.95	7.72	663.00	7.53

Table 3: EVPI and VSS performance Comparison of CPLEX stochastic whole model, CPLEX built-in Benders and our implemented Benders algorithm

		Benders-C		**Benders**		**MIP-full**	
$\|C\|$	$\|S\|$	EVPI	VSS	EVPI	VSS	EVPI	VSS
10	**20**	3.08	0.34	3.08	0.34	3.32	0.16
10	**40**	3.54	0.69	3.73	0.63	3.72	0.55
10	**60**	3.79	0.55	3.93	0.41	3.94	0.45
10	**80**	3.80	1.23	3.88	1.15	3.90	1.12
10	**100**	3.92	0.93	3.92	0.93	4.03	0.83

6.3 Analysis of Lifted Constraints

In Tables 4 and 5, we report the results produced by the MIP-full algorithm with and without the lifted formulation proposed in Section 4.3.1. For the experiment below, we abbreviate "MIP-full-L" for the CPLEX stochastic full model with the lifted formulation and "MIP-full-N" for the non-lifted alternative. UB, LB, and Gap as in the previous experiment are the best integer solution from branch-and-bound, the best bound, and the percentage difference, respectively. The Gap-diff refers to the value difference between the lifted and non-lifted versions, computed as $(\texttt{Gap}_{\text{MIP-full-N}} - \texttt{Gap}_{\text{MIP-full-L}})/\texttt{Gap}_{\text{MIP-full-L}}$ %. We set the computational time limit to be 1 hour and each row of the table is an average computed from 5 tests. We enabled the ALNS metaheuristic warm-start for both algorithms with and without the lifted formulation.

Table 4: Comparison of MIP-full-L and MIP-full-N algorithms

		MIP-full-L			MIP-full-N			
$\|C\|$	$\|S\|$	UB	LB	Gap	UB	LB	Gap	Gap-diff
20	20	305.33	203.52	33.34%	305.33	203.47	33.36%	0.05%
20	40	305.41	203.41	33.40%	305.41	203.33	33.42%	0.08%
20	60	306.10	237.21	22.51%	306.09	236.78	22.64%	0.61%
20	80	306.57	303.45	1.02%	306.60	303.34	1.06%	4.64%
20	100	307.32	303.38	1.28%	307.32	303.35	1.29%	0.76%
30	20	407.12	303.85	25.37%	407.10	303.67	25.41%	0.16%
30	40	408.80	303.60	25.73%	408.88	303.55	25.76%	0.10%
30	60	458.46	303.42	33.82%	458.46	303.24	33.86%	0.12%
30	80	508.16	303.22	40.33%	508.16	303.15	40.34%	0.03%
30	100	509.04	303.17	40.44%	509.04	303.13	40.45%	0.02%
40	20	509.56	405.40	20.44%	509.56	405.41	20.44%	-0.01%
40	40	576.44	405.21	29.71%	576.44	405.15	29.71%	0.03%
40	60	609.41	405.11	33.52%	609.41	405.09	33.53%	0.01%
40	80	609.73	404.98	33.58%	609.73	404.98	33.58%	0.00%
40	100	610.51	404.72	33.71%	610.51	404.70	33.71%	0.00%

From Table 4, we observe a better performance for MIP-full-L, reflected by a smaller UB-LB gap within the same computational time limit of an hour. MIP-full-L outperforms its non-lifted counterpart for almost all the instances apart from the $|C| = 40$, $|S| = 20$ case. For instances with a fixed customer size, MIP-full-L shows better effectiveness in finding stronger LB. This

Table 5: Comparison of MIP-full-L and MIP-full-N algorithms (with fixed number of integer nodes)

			MIP-full-L			MIP-full-N			
$\|C\|$	$\|S\|$	**nodes**	UB	LB	Gap	UB	LB	Gap	Gap-diff
20	20	100	305.22	203.97	33.17%	305.38	203.92	33.23%	0.16%
20	40	100	306.45	203.94	33.45%	306.43	203.74	33.51%	0.19%
20	60	100	306.89	203.88	33.56%	306.89	203.62	33.65%	0.26%
20	80	40	307.30	203.77	33.69%	307.30	203.53	33.77%	0.23%
20	100	15	307.69	203.10	33.99%	307.69	203.02	34.02%	0.08%
30	20	100	407.66	304.87	25.21%	407.66	304.45	25.32%	0.41%
30	40	100	408.99	304.62	25.52%	408.99	304.28	25.60%	0.33%
30	60	40	467.84	304.37	34.94%	467.84	304.19	34.98%	0.11%
30	80	10	508.94	304.10	40.25%	508.94	304.01	40.27%	0.04%
30	100	5	509.75	303.48	40.47%	509.75	303.41	40.48%	0.03%
40	20	10	509.97	405.42	20.50%	509.97	405.32	20.52%	0.10%
40	40	10	586.48	405.04	30.94%	586.48	404.95	30.95%	0.05%
40	60	10	609.76	404.94	33.59%	609.76	404.89	33.60%	0.03%
40	80	5	609.97	404.50	33.68%	609.97	404.47	33.69%	0.01%
40	100	3	610.85	404.27	33.82%	610.85	404.25	33.82%	0.01%

difference is the largest for the instance with $|C| = 20$ and $|S| = 80$. Nevertheless, we observe that a large number of lifted constraints is added to the MP that, on the one hand helps with locating the integer solution with higher quality in each iteration, whereas on the other hand, it can slow down the whole iterative convergence process by burdening the solving process with a more extensive set of constraints.

To cope with this, Table 5 gives another set of experimental results. We fix the same number of branch-and-bound nodes to be visited and compare the performances between MIP-full-L and MIP-full-N algorithms. The number of integer nodes is determined from the previous experiments so that both versions of the algorithm can explore the same number of branch-and-bound nodes within the time limit of an hour. On reaching the node limit, the algorithm will return the UB and LB. We observe a more extensive performance difference between MIP-full-L and MIP-full-N, especially for instances with many customers and scenarios. This shows our lifted formulation contributes to a faster convergence for the stochastic full model.

6.4 Two-stage Heuristic

In this section, we report the performance of our two-stage heuristic. For comparison, we include the following algorithms: "SVRP" is our stochastic H-SARA-2 model solved with the proposed time-saving root node approach that pre-fixes the fleet size m. "2-Stage Heur" is our two-stage heuristic method with the local search operators enabled in the ALNS improvement process. "2-Stage" ALNS is 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 and is regarded as the benchmark model in comparison to the two-stage heuristic. "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 customer cancellations cost the business

apart from other uncertainties.

The experiment results are given in Table 6, where all the results (Score and Time) are averaged from 10 experiments. Specifically, 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. Scenarios used in methods are generated randomly and independently from test instances. We have observed the following points: To begin with, our two-stage heuristic can tackle larger customer sizes 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 within 5 minutes, and within 10 minutes for 150 customers.

For the solution quality, our two-stage heuristic provides competitive solutions compared to CPLEX solutions. By comparing same-scenario columns between the exact methods and the 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 the global optimum is extremely difficult to compute), 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. For more than 10 customers, the "2-stage Heur" and "2-stage ALNS" columns show that our Overlap-breaker operator further improves the convergence speed and solution quality comparing to a classical ALNS heuristic.

Thus, we are able to include 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" (a priori decisions and previous-day customer notifications without getting the complete picture).

6.5 Analysis of Cancellation Policy

In this section, we design a second set of experiments to analyse the different cancellation types and their effect on the customers, service teams, and the decision-making process. We assume that the in-time cancellation policy allows sufficient time for rescheduling the service to prevent visits to the cancelled customers. In contrast, under the last-minute cancellation and no-presence policies, the service teams have to visit the cancelled customers, such that the arcs linking the cancelled nodes will be traversed and the service time at each cancelled node will be adjusted to 0 and 15 minutes, respectively. We fix a cancellation rate of 0.1 across all the following experiments.

From Figure 7, last-minute and no-presence cancellations result in less efficient schedules and higher idling time and overtime for the service teams. Since last-minute and no-presence cancellations occur only after the tour refinement stage in our two-stage decision process, the second-stage routes and schedules fail to satisfactorily adjust to the sudden change, which results in an additional idling time (at the consecutive customer nodes) of 25 minutes per service team if all customers cancel their service last minute. The no-presence scenario brings an additional 14 minutes of idling time to each service team on average, which is a 78% increase compared to the in-time cancellation scenario.

For instances with more than 30 customers, the last-minute and no-presence cancellations lead to a more prolonged service team overtime. Nevertheless, a longer idling time due to last-minute and no-presence cancellations can bring down the average waiting time for customers, since service teams arrive earlier than scheduled at the following customers. If all customers follow last-minute cancellation or no presence, the average overtime for each team will be 18 minutes more, which is a 40% increase, but the average customer waiting time will be 11 minutes less over all the experiments.

7 Conclusions

This paper studied a problem that integrates fleet-sizing, assignment, routing, and scheduling problems. A stochastic MIP model is proposed and it is solved using a customised Benders' decomposition algorithm. A closed-form solution is derived for the primal subproblem to allow the computational time to increase linearly with the instance size. A master-level lifted formulation was beneficial for reducing the number of feasibility cuts and accelerating the convergence of the algorithm. The adaptive large neighbourhood search from our two-stage heuristic is expanded to develop a metaheuristic-based warm-start process to speed up the convergence process. 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 computational time and solution quality performance. It also demonstrates 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 high-quality fleet sizing, districting, routing, and scheduling decisions with low idling, waiting, and overtime costs, as well as two sets of customer appointment time windows, and a balanced service team workload within geographically clear service zones.

Table 6: Computational results from different solution methods

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	
Customers n	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*	237.15	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

$*$ Notice that "2-stage" is used as an abbreviation to represent the "two-stage" in the heuristic experiment for space saving.

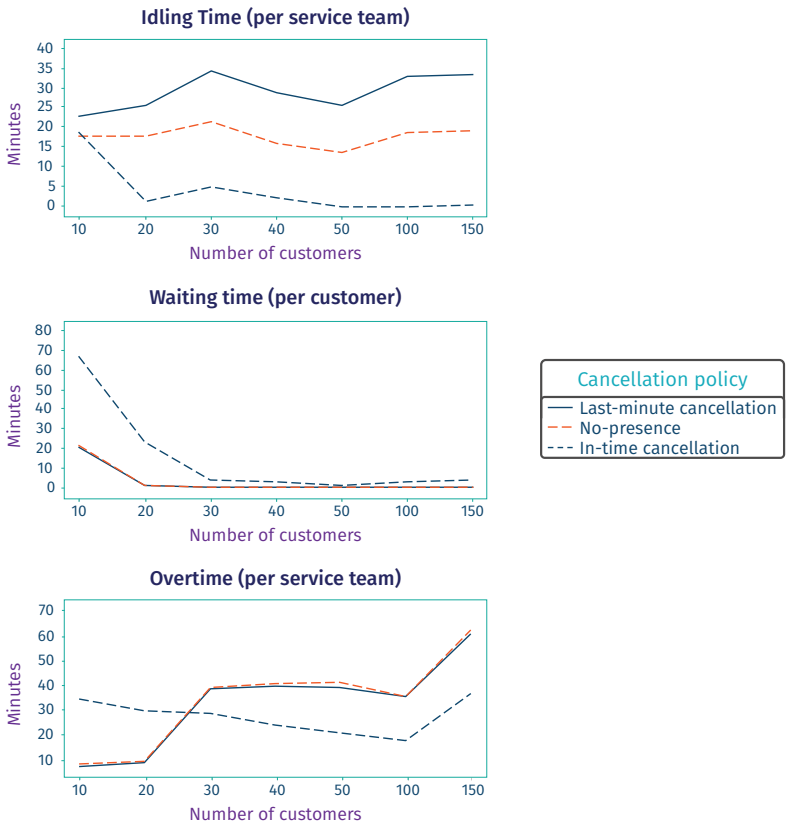


Figure 7: The idling time, waiting time and overtime from different cancellation policies

Acknowledgements. A.M-T. acknowledges support from the MAC–MIGS CDT Scholarship under EPSRC grant EP/S023291/1.

References

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

[Ben62] J. F. Benders. "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische Mathematik* 4.1 (1962), pp. 238–252.

[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.

[CLM18] G. Carello, E. Lanzarone, and S. Mattia. "Trade-off between stakeholders' goals in the home care nurse-to-patient assignment problem". In: *Operations Research for Health Care* 16 (2018), pp. 29–40.

[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.

[CF06] G. Codato and M. Fischetti. "Combinatorial Benders' cuts for mixed-integer linear programming". In: *Operations Research* 54.4 (2006), pp. 756–766.

[DFJ09] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. "Solution of a Large-Scale Traveling-Salesman Problem". In: *50 Years of Integer Programming 1958-2008*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 7–28.

[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.

[DMR14] G. Desaulniers, O. B. Madsen, and S. Ropke. "Chapter 5: The vehicle routing problem with time windows". In: *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. SIAM, 2014, pp. 119–159.

[DME21] M. Di Mascolo, C. Martinez, and M.-L. Espinouse. "Routing and scheduling in home health care: A literature survey and bibliometric analysis". In: *Computers & Industrial Engineering* (2021), p. 107255.

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

[EFR06] P. Eveborn, P. Flisberg, and M. Rönnqvist. "Laps Care—an operational system for staff planning of home care". In: *European Journal of Operational Research* 171.3 (2006), pp. 962–976.

[Eve+09] P. Eveborn, M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth. "Operations research improves quality and efficiency in home care". In: *Interfaces* 39.1 (2009), pp. 18–34.

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

[Fik+16] C. Fikar, A. A. Juan, E. Martinez, and P. Hirsch. "A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing". In: *European Journal of Industrial Engineering* 10.3 (2016), pp. 323–340.

[GR19] M. I. Gomes and T. R. P. Ramos. "Modelling and (re-) planning periodic home social care services with loyalty and non-loyalty features". In: *European Journal of Operational Research* 277.1 (2019), pp. 284–299.

[GUC20] L. Grieco, M. Utley, and S. Crowe. "Operational research applied to decisions in home health care: A systematic literature review". In: *Journal of the Operational Research Society* (2020), pp. 1–32.

[GD08] D. Gupta and B. Denton. "Appointment scheduling in health care: Challenges and opportunities". In: *IIE Transactions* 40.9 (2008), pp. 800–819.

[Gut+18] A. Gutierrez, L. Dieulle, N. Labadie, and N. Velasco. "A multi-population algorithm to solve the VRP with stochastic service and travel times". In: *Computers & Industrial Engineering* 125 (2018), pp. 144–156.

[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.

[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.

[HPR20] H. Hashemi Doulabi, G. Pesant, and L.-M. Rousseau. "Vehicle routing problems with synchronized visits and stochastic travel and service times: Applications in healthcare". In: *Transportation Science* 54.4 (2020), pp. 1053–1072.

[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.

[Hul+12] P. J. Hulshof, N. Kortbeek, R. J. Boucherie, E. W. Hans, and P. J. Bakker. "Taxonomic classification of planning decisions in health care: a structured review of the state of the art in OR/MS". In: *Health systems* 1.2 (2012), pp. 129–175.

[Joh+21] S.-N. Johnn, Y. Zhu, A. Miniguano-Trujillo, and A. Gupte. "Solving the Home Service Assignment, Routing, and Appointment Scheduling (H-SARA) Problem with Uncertainties". In: *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*. Ed. by M. Müller-Hannemann and F. Perea. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2021, 4:1–4:21.

[LMS10] E. Lanzarone, A. Matta, and G. Scaccabarozzi. "A patient stochastic model to support human resource planning in home care". In: *Production Planning and Control* 21.1 (2010), pp. 3–25.

[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.

[Lin+18] C.-C. Lin, L.-P. Hung, W.-Y. Liu, and M.-C. Tsai. "Jointly rostering, routing, and rerostering for home health care services: A harmony search approach with genetic, saturation, inheritance, and immigrant schemes". In: *Computers & Industrial Engineering* 115 (2018), pp. 151–166.

[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.

[McC76] G. P. McCormick. "Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems". In: *Mathematical Programming* 10.1 (1976), pp. 147–175.

[Nad+23] B. Naderi, M. A. Begen, G. S. Zaric, and V. Roshanaei. "A novel and efficient exact technique for integrated staffing, assignment, routing, and scheduling of home care services under uncertainty". In: *Omega* 116 (2023), p. 102805.

[ND18] J. A. Nasir and C. Dang. "Solving a more flexible home health care scheduling and routing problem with joint patient and nursing staff selection". In: *Sustainability* 10.1 (2018), p. 148.

[NSS12] S. Nickel, M. Schröder, and J. Steeg. "Mid-term and short-term planning support for home health care services". In: *European Journal of Operational Research* 219.3 (2012), pp. 574–587.

[NBA21] E. Nikzad, M. Bashiri, and B. Abbasi. "A matheuristic algorithm for stochastic home health care planning". In: *European Journal of Operational Research* 288.3 (2021), pp. 753–774.

[Rah+17] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. "The Benders decomposition algorithm: A literature review". In: *European Journal of Operational Research* 259.3 (2017), pp. 801–817.

[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*. European Journal of Operational Research 211.3 (2011), pp. 427–441.

[RH16] K.-D. Rest and P. Hirsch. "Daily scheduling of home health care services using time-dependent public transport". In: *Flexible Services and Manufacturing Journal* 28.3 (2016), pp. 495–525.

[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.

[SI10] G. K. Saharidis and M. G. Ierapetritou. "Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy". In: *Computers & Chemical Engineering* 34.8 (2010), pp. 1237–1245.

[Sha+21] S. Shahnejat-Bushehri, R. Tavakkoli-Moghaddam, M. Boronoos, and A. Ghasemkhani. "A robust home health care routing-scheduling problem with temporal dependencies under uncertainty". In: *Expert Systems with Applications* 182 (2021), p. 115209.

[Sha99] P. Shaw. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In: *Principles and Practice of Constraint Programming — CP98*. 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.

[SBG17] Y. Shi, T. Boudouh, and O. Grunder. "A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand". In: *Expert Systems with Applications* 72 (2017), pp. 160–176.

[SBG19] Y. Shi, T. Boudouh, and O. Grunder. "A robust optimization for a home health care routing and scheduling problem with consideration of uncertain travel and service times". In: *Transportation Research. Part E: Logistics and Transportation Review* 128 (2019), pp. 52–95.

[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.

[TGH11] A. Trautsamwieser, M. Gronalt, and P. Hirsch. "Securing home health care in times of natural disasters". In: *OR spectrum* 33.3 (2011), pp. 787–813.

[YNY21] M. Yang, Y. Ni, and L. Yang. "A multi-objective consistent home healthcare routing and scheduling problem in an uncertain environment". In: *Computers & Industrial Engineering* 160 (2021), p. 107560.

[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* 53.24 (2015), pp. 7450–7464.

[YLJ18] B. Yuan, R. Liu, and Z. Jiang. "Daily scheduling of caregivers with stochastic times". In: *International Journal of Production Research* 56.9 (2018), pp. 3245–3261.

[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.

A APPENDIX

Table A.1: HHC papers with a focus on routing and scheduling and uncertainties

Entry		HHC Decisions				Uncertainty			Solution Method	
Author	**Year**	**Fleetsize**	**Assignment**	**Routing**	**Scheduling**	**Travel Time**	**Service Time**	**Cancellation**	**Modelling Approach**	**Solution Approach**
Begur	1997		*	*	*				MIP	Multiple heuristics
Cheng	1997		*	*	*				MIP	Two-phase heuristic
De Angelis	1998		*				*		Stochastic LP	NA
Blais	2003		*						MIP	Tabu Search
Bertels	2006		*	*	*				MIP	Combination of constraint programming, tabu search and simulated annealing
Eveborn	2006		*	*	*			*	IP, Set Partitioning	Repeated matching algorithm
Akjiratikarl	2007		*	*	*				IP	Particle Swarm Optimisation Meta-heuristic
Eveborn	2009		*	*	*			*	IP, Set Partitioning	Repeated matching algorithm
Hertz	2009		*						MIP	Commercial solver (CPLEX), Meta-heuristic based on Tabu Search
Bennett	2011			*	*				Combinatorial optimisation	Greedy heuristic
Trautsamwieser	2011		*	*	*			*	IP	Meta-heuristic (VNS)
Trautsamwieser	2011		*	*	*				IP	Meta-heuristic (VNS)
An	2012			*	*				MIP	Two-phase heuristic
Bennett	2012		*	*	*				MIP	Evaluations
Koeleman	2012	*	*	*	*				Markov decision process	Heuristic
Lanzarone	2012		*				*		Stochastic Programming	Structural policy
Lanzarone	2012		*				*		(Stochastic) Integer programming	Not specified - OPL 5.1 solver used
Nickel	2012		*	*	*			*	Constraint programming	two-phase method based on Constraint Programming and ALNS metaheuristic
Rasmussen	2012		*	*	*				IP (Set Partitioning formulation)	Branch-and-Price
Shao	2012		*	*	*				MIP	Greedy randomised adaptive search procedure (GRASP)
Allaoua	2013	*	*	*	*				ILP	decomposition-based heuristic
Bard	2013		*	*					MIP	Commercial solver and Heuristics
Benzarti	2013		*						MIP	Commercial solver (CPLEX)
Liu	2013		*	*	*				MIP	tabu search and genetic algorithm
Bard	2014		*	*	*				MIP	Heuristic based on a greedy randomised adaptive search procedure (GRASP)
Bard	2014		*	*	*				MIP	Branch-cut-and-price, rolling horizon heuristic
Carello	2014		*				*		Robust Optimisation	Commercial solver (CPLEX)
Kergosien	2014		*	*	*				MIP	Tabu-search + Variable neighbourhood search
Lanzarone	2014		*				*		Analytical	Analytical policy
Liu	2014		*	*	*				MIP	Tabu Search
Mankowska	2014		*	*	*				MIP	Heuristic
Trautsamwieser	2014		*	*	*				MIP	Branch-and-Price-and-Cut
Yalçındağ	2014		*	*		*			MIP, Kernel Regression	Assignment-first-routing-second and simutaneous policies
Bowers	2015		*	*	*				Combinatorial optimisation	Modified Clarke-Wright algorithm, simulation
Cappanera	2015		*	*	*				ILP	Branch and Bound
Fikar	2015		*	*	*				IP	two-stage matheuristic based on tabu search
Hiermann	2015		*	*	*				Constraint programming	Two-stage heuristic
Maya	2015		*	*	*				Bi-objective MIP	Three-stage heuristic
Nguyen	2015		*	*	*			*	Robust Optimisation	Matheuristic
Rodriguez	2015	*					*		Stochastic Proramming	Branch-and-cut
Rest	2015		*	*	*				MIP	Meta-heuristic approach based on tabu search Tabu search-based metaheuristic
Yuan	2015		*	*	*		*		Stochastic Programming	Branch-and-price
Braekers	2016		*	*	*				MIP	Meta-heuristic
Decerle	2016		*	*	*				MIP	Two-phase metaheuristic
Errarhout	2016		*				*		Stochastic Programming	Commercial solver (CPLEX)
Fikar	2016		*	*	*			*	IP	Discrete-event driven metaheuristic
Lin	2016		*						MIP	Commercial solver (Gurobi)
López	2016		*	*	*			*	Multi-agent approach	Commercial solvers (CPLEX, JADE)
Redjem	2016			*	*				MIP	Heuristic Approach

Table A.1 (continued)

Entry		HHC Decisions				Uncertainty			Solution Method	
Author	**Year**	**Fleetsize**	**Assignment**	**Routing**	**Scheduling**	**Travel Time**	**Service Time**	**Cancellation**	**Modelling Approach**	**Solution Approach**
Rest	2016		*	*	*	*			MIP	Combination of dynamic programming and meta-heuristics based on tabu search
Wirnitzer	2016		*		*				MIP	Commercial solver (Gurobi)
Yalçındağ	2016		*	*	*				ILP	Patter-based two-phase approach
Du	2017		*	*	*				MIP	Genetic algorithm
Guericke	2017		*	*	*				MIP	Meta-heuristic approach based on adaptive large neighbourhood search (ALNS)
Han	2017		*	*	*		*	*	MIP	hybrid heuristic algorithm (DP + TS)
Laesanklang	2017		*	*	*				MIP	Heuristic decomposition
Lin	2017		*						MIP	Greedy heuristic
Liu	2017		*	*	*				MIP	Branch-and-price
Shi	2017		*	*	*		*		MIP + fuzzy chance constraint	Hybrid genetic algorithm
Shi	2017		*	*	*	*	*		Stochastic Programming	Hybrid genetic algorithm
Yuan	2017			a priori	a priori			*	IP	Heurisitic
Cappanera	2018		*	*	*			*	Robust Optimisation	Metaheuristic
Carello	2018		*				*		MIP	Commercial solver (CPLEX)
Decerle	2018		*	*	*				MIP	Memetic Algorithm
Issabakhsh	2018		*	*	*	*			MIP + Robust optimisation	uncertainty level analysis
Lin	2018		*	*	*			*	MIP	Heuristic
Liu	2018		*	*	*				Bi-objective MIP	Metaheuristic
Nasir	2018	*	*	*	*				MIP	heuristic algorithm based on VNS approach
Nasir	2018	*	*	*	*				ILP	Commercial solver (CPLEX)
Shi	2018		*	*	*	*	*		Stochastic Programming	Commercial solver (CPLEX) & SA-based heuristic algorithm
Yuan	2018		*	*	*	*	*		Stochastic Programming	Branch-and-Price
Zhan	2018		*	*	*		*		Stochastic Programming, MIP	Tabu Search-based heuristic
Demirbilek	2019		*	*	*			*	MIP	Scenario Based heuristic Approach
Grenouilleau	2019		*	*	*				MIP	set partitioning heuristic
Heching	2019		*		*				MIP, constraint programming	Logic-based Benders Decomposition, Branch-and-check
Liu	2019		*	*	*	*	*		IP	Branch-and-Price
Shi	2019		*	*	*	*	*		Robust Optimisation	Gurobi Solver, Heuristics (Simulated Annealing, Tabu Search, and Variable Neighbourhood Search)
Frifita	2020		*	*	*				MIP	Variable Neighbourhood Search
Grenouilleau	2020		*		*				Two-stage formulation	Logic-based Benders Decomposition, LNS meta-heuristic
Kandakoglu	2020		*	*	*				MIP	Specific decision support system (HDSS)
Restrepo	2020	*	*		*		*		Stochastic Programming	Commercial solver (CPLEX)
Shahedah	2020	*	*	*	*		*		Robust Optimisation	decomposition-based algorithm
Zhan	2020		*	*	*		*		MIP	L-shapted Method, MTSP heuristic
Bazirha	2021		*	*	*	*	*		Stochastic Programming with Recourse	Commercial solver (CPLEX) , genetic algorithm (GA), general variable neighbourhood search (GVNS)
Carello	2021		*				*		MIP, Robust Optimisation	Commercial solver (CPLEX)
Cinar	2021		*	*	*				MIP	Adaptive Large Neighbourhood Search (ALNS) and iterative mathematical programming
Demirbilek	2021		*	*	*			*	MIP	Scenario Based heuristic Approach
Khodabandeh	2021		*	*	*				Bi-objective optimisation	Epsilon-constraint-based approach
Li	2021		*	*	*				nonlinear & convex programming	"outer-approximation method, genetic algorithm "
Liu	2021		*	*	*				MIP	Matheuristic integrating (adaptive large Neighbourhood search, ALNS) heuristic and commercial solver (Gurobi)
Liu	2021		*	*	*				MIP	Hybrid metaheuristics
Naderi	2021	*	*	*	*	*	*		MIP	Benders Decomposition (logic-based Benders branching-decomposition algorithm)
Nikzad	2021	*	*	*	*	*	*		Stochastic Programming	Multi (two)-phase matheuristic
Bushehri	2021		*	*	*	*	*		robust Optimisation	Meta-heuristic (simulated annealing, genetic algorithm, memetic algorithm)
Shiri	2021		*	*	*		*		Robust Optimisation	Hybrid three-phase procedure
Xiang	2021		*	*	*				Bi-objective MIP	Hybrid elitist nondominated sorting genetic algorithm (hybrid NSGA-II)
Yang	2021		*	*	*	*	*		Uncertain Programming	Metaheuristic
Yadav	2022		*	*	*				MIP	Metaheuristic
Ours	**2023**	*	*	*	*	*	*	*	MIP	**Benders Decomposition, two-stage heuristic**

Table A.2: Table of symbols (in alphabetical order)

Symbol	Definition	Description
a_i	Service team's arrival time at customer i	Continuous variable
A	Arc set enclosing all edges linking any two nodes	Non-negative integer
AM_i	Activity measure for customer i	Non-negative parameter
$\mathsf{AM}_{\min}$	Minimum percentage of activity measure in a district	Non-negative parameter
$\mathsf{AM}_{\max}$	Minimum percentage of activity measure in a district	Non-negative parameter
b_i	Number of closest customers to i	Positive integer
c_{wait}	Customer waiting cost (per unit time)	Non-negative parameter
c_{idle}	Team idling cost (per unit time)	Non-negative parameter
c_{over}	Team overtime cost (per unit time)	Non-negative parameter
$d_{i,j}$	Travel (Euclidean) distance between i and j	Non-negative parameter
e_i	Expected travel time from customer i to j (same cluster)	Positive number
f_m	Cost for hiring any service team m	Positive parameter
g_i	Overtime length returning to depot from customer i	Continuous variable
h_i	Idle time of service team at customer i	Continuous variable
$\mathbb{I}$	Set of customers with service cancellations	Non-negative integer set
I	Customers set	Non-negative integer set
i, j	Node index	Non-negative integer
ℓ_l	Lower bound on the number of teams	Non-negative integer
ℓ_u	Upper bound on the number of teams	Non-negative integer
L	Standard maximum allowed working time	Positive parameter
L_δ	Safety time at the end of each tour	Positive parameter
m	Service team index	Non-negative integer
$\widehat{m}$	Upper limit for the service team	Non-negative parameter
M	Big M	Positive integer
n	Number of customers	Non-negative parameter
q_i	cancellation probability of customer i	Non-negative parameter
$p_{i,j}^{(*)}$	Likelihood of customer j following i on a route	Non-negative number
q_ω	Scenario ω's associated probability	non-negative parameter
$R_{i,j}$	Rank of the jth closest customer to i,	Positive integer
s_i^ω	Service time at customer i under scenario ω	Positive parameter
$\widehat{s}_i$	Expected service time at customer i	Positive number
S	Stochastic service duration vector	non-negative parameter
t_i	Appointment time of customer i	Continuous variable
$\tau_{i,j}^\omega$	Traversal time from node i to j under scenario ω	Positive parameter
$\widehat{\tau}_{i,j}$	Expected traversal time from node i to j	Positive number
T	Stochastic traversal duration matrix	non-negative parameter
T_1	First-stage customer appointment time window (length)	Positive parameter
T_2	Second-stage customer appointment time window (length)	Positive parameter
$v_{i,j}$	Traversal speed from node i to j	Positive number
$\widehat{v}_{i,j}$	Expected travelling speed from node i to j	Positive number
V	Node set enclosing all customer nodes	Non-negative integer
w_i	Waiting time for customer i	Continuous variable
W	Appointment time window half width	Continuous variable
$x_{i,j}$	Determines if arc (i, j) is traversed by a team	Integer (binary) variable
$y_{i,j}$	If customer i is assigned to district centred at j	Integer (binary) variable
ω	Scenario with travel and service times realisations	samples
Ω	Total set of scenarios $[\omega]$	samples set
θ	Maximum overtime length	Non negative parameter
λ	Unit travel time cost (per hour)	Positive parameter
μ	Average activity measure per district	Non-negative number

The notations from ALNS heuristic in Section 5.3.4 are explained as part of the ALNS methodology and hence not listed.