


A QUEUEING-BASED APPROACH FOR INTEGRATED ROUTING AND APPOINTMENT SCHEDULING

RENÉ BEKKER, BHARTI BHARTI, LEON LAN, MICHEL MANDJES

ABSTRACT. This paper aims to address the integrated routing and appointment scheduling (RAS) problem for a single service provider. The RAS problem is an operational challenge faced by operators that provide services requiring home attendance, such as grocery delivery, home healthcare, or maintenance services. While considering the inherently random nature of service and travel times, the goal is to minimize a weighted sum of the operator’s travel times and idle time, and the client’s waiting times. To handle the complex search space of routing and appointment scheduling decisions, we propose a queueing-based approach to effectively deal with the appointment scheduling decisions. We use two well-known approximations from queueing theory: first, we use an approach based on phase-type distributions to accurately approximate the objective function, and second, we use an heavy-traffic approximation to derive an efficient procedure to obtain good appointment schedules. Combining these two approaches results in a fast and sufficiently accurate hybrid approximation, thus essentially reducing RAS to a routing problem. Moreover, we propose the use a simple yet effective large neighborhood search metaheuristic to explore the space of routing decisions. The effectiveness of our proposed methodology is tested on benchmark instances with up to 40 clients, demonstrating an efficient and accurate methodology for integrated routing and appointment scheduling.

KEYWORDS. Appointment scheduling ◦ Routing ◦ Phase-type distributions ◦ Heavy-traffic ◦ Large neighborhood search

AFFILIATIONS. *Bharti Bharti* and *Michel Mandjes* are with the Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, the Netherlands. *MM* is also with Mathematical Institute, Leiden University, The Netherlands, and Amsterdam Business School, Faculty of Economics and Business, University of Amsterdam, Amsterdam, the Netherlands. This research was supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 945045, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003. 

René Bekker and *Leon Lan* are with the Department of Mathematics, Vrije Universiteit, De Boelelaan 1111, 1081 HV Amsterdam, the Netherlands.

1. INTRODUCTION

The market for providing services at home is growing rapidly and encompasses a wide range of services, such as food delivery, home healthcare and wellness, maintenance and appliance repair, household activities, and more. A specific challenge for home attendance services is the combination of an efficient home service process combined with good quality of service (e.g. measured as ‘right on time’ home attendance). Hence, the operational challenge that the service provider is facing lies in determining, for a given set of clients with known locations, the corresponding route and appointment times. This combined routing and appointment scheduling problem has become a critical issue, as inefficient scheduling and sub-optimal resource allocation can have significant cost implications. Studies [33] indicate that these inefficiencies cost the US healthcare system more than \$150 billion annually, resulting in reduced provider productivity, delayed access to care, and increased burden on the healthcare system. Similarly, in the transportation sector, improper scheduling and routing have been associated with increased fuel consumption, carbon emissions, and total logistics delivery costs [26]. The significance of this issue extends beyond financial aspects, encompassing enhanced service quality, improved client satisfaction, and the increasing pressure and competition faced by delivery companies to ensure on-time performance [4]. Therefore, it is important to study this problem, so as to develop effective scheduling strategies and solutions.

The main objective of this paper is to develop a unified approach to solve the integrated routing and appointment scheduling (RAS) problem; an illustration of the RAS problem is depicted in Figure 2. As opposed to what is usually considered in the literature, the setup we consider is intrinsically random. More precisely, to do justice to their inherently random nature, we model the service times (the times spent at the clients’ locations) and travel times (the time it takes to travel between two subsequent locations) as random variables. The service provider is required to visit a predetermined set of locations before returning to their initial location, e.g., a warehouse. The clients residing at these locations are provided with appointment times.

Importantly, the service provider and the clients have conflicting interests. The service provider wishes to minimize their travel time and, in addition, they want to avoid arriving at the client’s location before the corresponding appointment time resulting in idle time of the service provider. Instead, the clients prefer short waiting times. Therefore, in our approach, we aim to minimize these three elements so as to strike a proper balance between the interests of both the service provider and the clients. Even in a setting without any randomness, this problem is intrinsically hard: the routing decision leads to the traveling salesman problem (TSP), which in itself is an NP-hard problem. A major challenge lies in the randomness that we wish to incorporate: it is not evident how to accurately and efficiently evaluate the objective function for given distributions of travel times and service times.

Problem structure, benchmark algorithms. Routing and appointment scheduling are mature subfields of operations research, in each of which a wealth of well-performing algorithms is available. In appointment scheduling, an excellent heuristic for the ordering of clients is the so-called *smallest variance first* (SVF) rule, which sequences clients in order of increasing variance of their service durations. While this rule is technically not optimal [1], it has been formally proven that in specific asymptotic regimes it *is* [6]. The informal backing for SVF’s near-optimality lies in the fact that one wishes to introduce as little as possible uncertainty into the system. If one schedules relatively deterministic jobs first, they hardly do any harm to their successors.

The intrinsic difficulty of the RAS problem lies in the fact that it is not straightforward to combine algorithms from appointment scheduling and routing (TSP), as can be seen as follows. A naïve method would be to replace the travel times and service times by their respective means, and to use off-the-shelf techniques to solve the resulting TSP (‘mean based TSP’, that is). The problem with this approach,

however, is that the clients early in the tour could be the ones corresponding to a large variance, thus contradicting the principles underlying SVF. Informally, one could say that the solution to the RAS problem finds a compromise between TSP and SVF. This effect is illustrated by Figure 1: it shows that in this instance TSP leads to a route that differs strongly from the SVF-based one, while the optimal route (determined using full enumeration) combines elements of both.

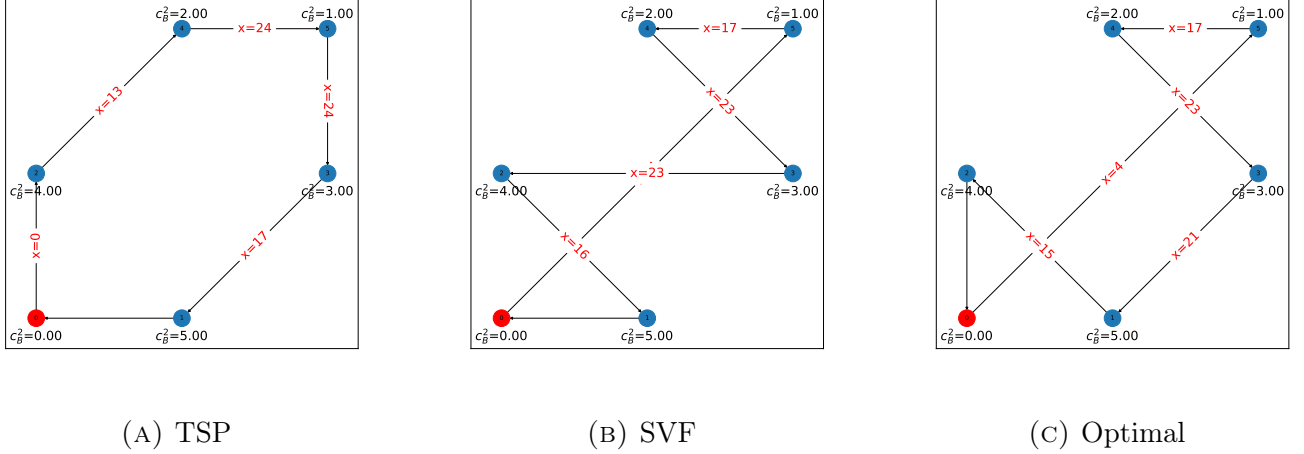


FIGURE 1. Comparison of solutions obtained by the (mean-based) TSP and SVF algorithms, and the optimal solution. The red node denotes the depot, while the blue nodes correspond to clients. The node attributes represent the SCV of each node, which is to be interpreted as a measure for the variability of the corresponding service time; mean service times equal 1 and travel times are assumed deterministic. The edge attributes display the optimal inter-appointment times calculated for the corresponding tour.

A standard method to incorporate randomness is through the *sample average approximation* approach [19, 29], which involves repeatedly sampling random variables to create a large number of scenarios. Generally, a mixed-integer linear program (MILP), which includes both routing and appointment scheduling decisions, is solved across all these scenarios. Solving such MILPs with off-the-shelf optimization software is ineffective, but tailor-made solutions, such as the L-shaped method, make it possible to optimally solve instances with up to ten clients [43]. However, dealing with instances beyond this size remains highly challenging.

Contributions. Our approach radically differs from earlier studies. Rather than working with sampled scenarios, we incorporate the randomness into the objective function, which is subsequently optimized using soundly tuned metaheuristics. In more detail, our contributions are the following:

- We cast the objective function, for a given route and given appointment times, in terms of an appropriately chosen *queueing model*. In this queueing model, the arrival times are deterministic, whereas we represent the service times by random quantities with *phase-type distributions*. This class of distributions combines attractive properties: it is capable of closely approximating any given distribution [2], but at the same time it allows a high level of mathematical tractability. Building on the ideas of [9, 16], we are thus able to establish an explicit expression for our objective function.
- Due to the large matrices involved in the phase-type method described above, the evaluation of the objective function becomes prohibitively slow for problem instances of realistic size. We

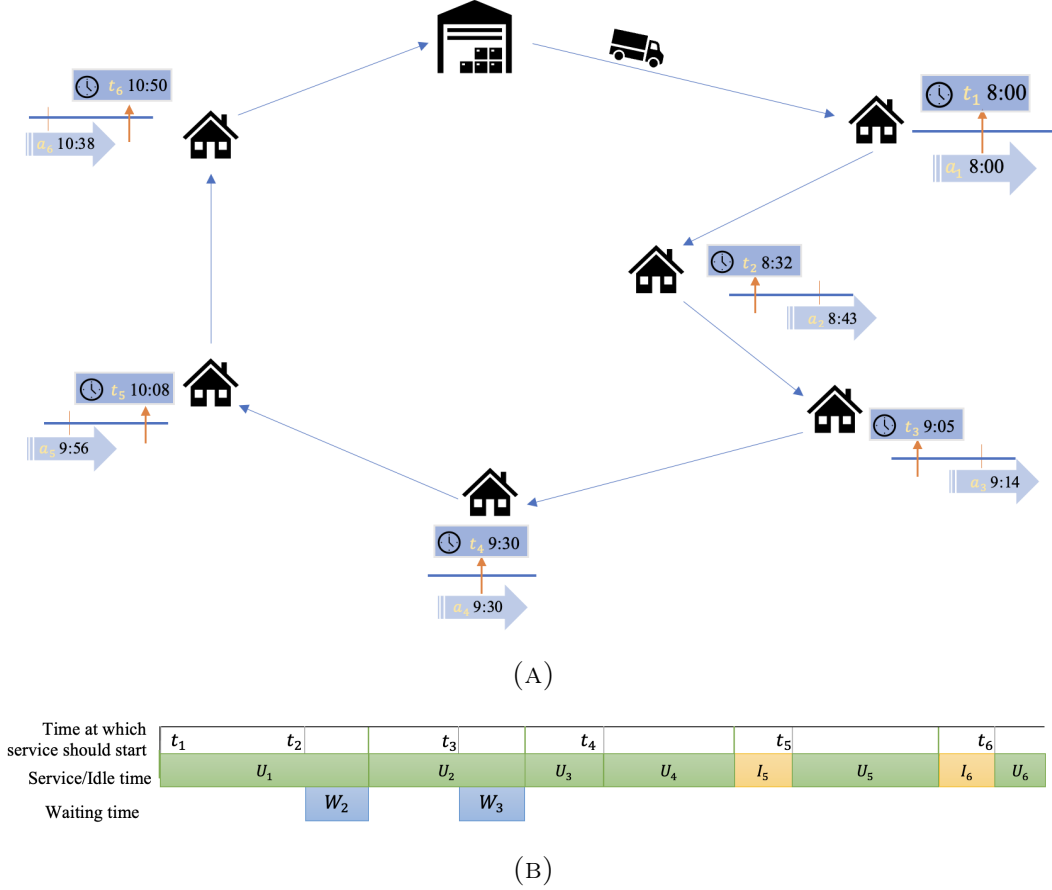


FIGURE 2. (A) shows a service provider with a route consisting of six locations. Here, t_j and a_j are the appointment time and arrival time at the j^{th} location, respectively. (B) shows the relation between ‘extended service time’ U_j , idle time I_j of the service provider before service of j^{th} client can start, and waiting time W_j of the j^{th} client.

remedy this by applying a *heavy-traffic approximation*, which provides an easy-to-compute, closed-form expression for the objective function. This expression allows explicit optimization over the appointment times (for a given route, that is). While the heavy-traffic approximation is not necessarily precise, it succeeds in capturing the essence of the underlying dynamics. To have the best of both worlds, we use a hybrid approach for determining the optimal route in our algorithm.

- The optimal route is found relying on the well-developed and well-tested body of algorithms known as *metaheuristics* [11]. More concretely, we implement a *large neighborhood search* metaheuristic [27], which improves a tour by iteratively destroying and repairing parts of the route. Combined with the fast heavy-traffic approximation, our proposed technique can handle instances of up to 40 clients within a reasonable amount of time. Through comprehensive numerical experiments we show that our approach improves over other algorithms, including the modified TSP heuristic from [43].

Organization. This paper is organized as follows. In Section 2 we provide an account of the relevant literature, both on the routing and the appointment scheduling component. Section 3 formally defines the RAS problem. Then, Section 4 provides a closed-form expression by which the objective function can be evaluated, as well as its explicit heavy-traffic approximation. It also demonstrates that the heavy-traffic approximation captures the essence of the underlying random dynamics. The large neighborhood search

metaheuristic is presented in Section 5. Section 6 presents a large set of numerical experiments. Finally, Section 7 concludes the paper.

2. LITERATURE

In this section, we consider some related literature concerning appointment scheduling (AS), routing, and the combined routing and appointment scheduling (RAS).

As mentioned, appointment scheduling is one of the two aspects of RAS problems. In particular, when we fix the route and focus on the appointment times, then our problem is reduced to AS. There is a long tradition of AS problems, initiated by the seminal paper of Bailey and Welch [39] who consider an appointment system for a hospital's outpatient department. As the performance analysis of an AS system is clearly non-trivial, some early papers have evaluated different appointment rules using simulation [10, 14]. However, the majority of papers focus on finding optimal appointment times. For example, [8] use a sequential bounding approach to frame the problem as a linear program and solve it using an L -shaped method. Some other approaches to finding optimal appointment times are, for instance, simulation optimization [17, 18] and methods based on local search [16, 41]. In fact, [41] show that the AS problem for discrete appointment epochs is multimodular, allowing for an efficient optimization approach.

Another stream of research employs phase-type approximations for the service times in AS problems. Phase-type distributions can approximate any distribution arbitrarily closely and they allow for exact results using matrix algebraic manipulations; see [36, 38] for some early studies. In the current paper, we mainly follow the approach of [22, 25] for an exact evaluation. In those studies, the authors fit a phase-type distribution to the service times based on the first two moments. The optimal appointment times are determined numerically by exploiting the convexity of the objective function [25], which is a weighted combination of idle and waiting times in the AS setting. Another approach to determine optimized appointment times is to use stationary models and/or heavy traffic approximations; such approximations have been applied in [24] for the single-server case and in [23] for the multi-server situation. We refer to [1, 2, 5, 12] for some review papers about AS, which are all motivated by applications in health care.

The second aspect of our RAS problem is the optimization of the route. If we neglect the waiting and idle times at the nodes and only consider travel time, then the problem reduces to a TSP, which is closely related to the class of vehicle routing problems (VRPs). Both the TSP and VRP are classical problems in combinatorial optimization; see e.g. [6] for a historic treatment of TSP and [20] for a recent survey on VRP. Many heuristics and metaheuristics have been developed to solve VRPs, of which large neighborhood search (LNS) has been very successful. We refer to [27] for some applications of LNS to variants of TSP and VRP.

Moreover, there is a limited amount but a variety of literature concerning combined routing and appointment scheduling. For instance, a stream of papers considers VRP models with time window assignment, in which a time window for each client needs to be determined before demand becomes known [7, 32]. Typically, these models focus on minimizing travel and/or operator costs and impose hard time window constraints. Other studies focus on robust optimization for VRP problems with time windows under uncertainty [15, 31]. Furthermore, [42] considers a stochastic programming approach in which flexible appointment times are allowed with a penalty on idle and waiting time. These papers incorporate uncertainty in their service times using scenarios that stem from the stochastic and robust programming domains, which differs from the way in which we incorporate uncertainty in our model. A difficulty with a scenario-based approach is that if the number of nodes becomes larger, it will be more challenging to represent the joint service times by representative sets of scenarios. In addition, the size of stochastic programs grows with the number of scenarios, posing computational challenges [28]. Instead, the basics of our approach stem from queueing-theoretic dynamics, as in AS problems.

Finally, the two papers that consider models that are most closely related to our model are [35, 43]. The RAS problem in [43] is similar to ours, albeit that we also consider stochastic travel times. Again, the fundamental difference is the use of scenarios to represent uncertainty in service times. The authors of [43] develop a MILP and an L -shaped method next to an easy-to-implement heuristic. In the experiments, they manage to handle instances of up to 10 nodes. The model in [35] is similar, but the authors additionally consider two distributionally robust optimization models. Similar to the results presented in [43], their experiments have up to 10 clients. Our work shows that the application of appropriate queueing models within the optimization allows us to consider larger instances.

3. MODEL FORMULATION

In this section, we formulate the integrated routing and appointment scheduling (RAS) problem. We consider a single operator situated at a starting location denoted by the index 0. The operator has to make a tour along n client locations, indexed by $1, \dots, n$, after which it returns to its starting location. The travel time between two locations $i, j = 0, 1, \dots, n$ with $i \neq j$ is denoted by T_{ij} and the service time at client i is denoted by B_i . Both T_{ij} and B_i are assumed to be non-negative, independent and identically distributed random variables. A special case of specific interest is when T_{ij} and B_i follow a phase-type distribution.

We define a tour $\sigma = (\sigma_1, \dots, \sigma_n)$ as a sequence of client indices, where σ_j denotes the index of the j -th visited client location. For notational convenience, we assume $\sigma_0 = \sigma_{n+1} = 0$, i.e., the starting location. For a given tour σ , the operator assigns to each client σ_j an appointment time $t_{\sigma_j} \geq 0$ at which the operator plans to provide service to the client, and we assume that $t_0 = 0$. The inter-appointment time $x_{\sigma_j} := t_{\sigma_j} - t_{\sigma_{j-1}}$ denotes the time between the appointment times of clients σ_{j-1} and σ_j .

When the operator arrives at a client earlier than scheduled, we say that there is *idle time*, whereas if the operator arrives later than scheduled, we say there is *waiting time*. Throughout, we denote by I_{σ_j} the idle time prior to the appointment time at client σ_j , whereas W_{σ_j} is the waiting time of client σ_j ; see part (B) of Figure 2 for a schematic representation. With slight abuse of notation, we denote by $T_{\sigma_j} := T_{\sigma_{j-1}, \sigma_j}$ the travel time from location σ_{j-1} to location σ_j . For $j = 1, \dots, n$, the following recurrence relations apply:

$$\begin{aligned} I_{\sigma_j} &= \max\{x_{\sigma_j} - T_{\sigma_j} - W_{\sigma_{j-1}} - B_{\sigma_{j-1}}, 0\}, \\ W_{\sigma_j} &= \max\{T_{\sigma_j} + W_{\sigma_{j-1}} + B_{\sigma_{j-1}} - x_{\sigma_j}, 0\}, \end{aligned}$$

with $W_0 = 0$ and $B_0 = 0$. In queueing theory, these relations are known as the *Lindley recursion* [3, Section III.6]. They define a queueing model in which the inter-arrival times are given by the x_{σ_j} and the service requirements by the random variables $U_{\sigma_j} := T_{\sigma_j} + B_{\sigma_{j-1}}$. Note that $U_{\sigma_1} = T_{\sigma_1}$, i.e., U_{σ_1} equals the travel time from the depot to the first location σ_1 and does not include any service time.

Our goal is to find a tour $\sigma \equiv (\sigma_1, \dots, \sigma_n)$ and inter-appointment times $\mathbf{x} \equiv (x_{\sigma_1}, \dots, x_{\sigma_n})$ that minimize an objective function that is a weighted sum of three elements: (i) the expected value of the total travel time of the tour, (ii) the expected aggregate idle times of the operator, and (iii) the expected waiting times of the individual clients.

Hence, our objective function is, for weights $\omega^T, \omega^I, \omega_j^W \geq 0$,

$$\mathcal{L}(\sigma, \mathbf{x}) = \omega^T \sum_{j=1}^{n+1} \mathbb{E} T_{\sigma_j} + \omega^I \sum_{j=1}^n \mathbb{E} I_{\sigma_j} + \sum_{j=1}^n \omega_{\sigma_j}^W \mathbb{E} W_{\sigma_j}. \quad (1)$$

Note that we allow clients to have heterogeneous waiting time weights ω_j^W . Moreover, observe that the quantities $\mathbb{E} T_{\sigma_j}$ depend on the ordering σ , whereas the quantities $\mathbb{E} I_{\sigma_j}$ and $\mathbb{E} W_{\sigma_j}$ depend on both the

ordering σ and the inter-appointment times \mathbf{x} . The RAS problem is thus defined as

$$\mathcal{L}^* = \min_{\sigma \in \mathcal{S}, \mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma, \mathbf{x}), \quad (\text{RAS})$$

where \mathcal{S} is the collection of permutations of $\{1, \dots, n\}$.

The RAS problem is challenging for two main reasons. First, it requires the exploration of a complex, integrated search space of routing and appointment scheduling decisions, where the latter depends on the chosen tour. Second, the evaluation of $\mathcal{L}(\sigma, \mathbf{x})$ for a given ordering σ and given inter-appointment times \mathbf{x} requires the computation of expected values of the idle times I_j and the waiting times W_j , which is effectively equivalent to computing their full distributions. This is, in general, problematic: for general travel and service time distributions, one cannot obtain closed-form expressions for the density or distribution function of the idle and waiting times. In the next section, we show that there are accurate and effective approximation procedures from queueing theory that can provide closed-form expressions.

4. QUEUEING APPROXIMATIONS FOR THE SCHEDULING PROBLEM

In this section, we consider the appointment scheduling part of the RAS problem. We present two queueing-based approximation procedures to evaluate the objective function $\mathcal{L}(\sigma, \mathbf{x})$ for a given tour σ and inter-appointment times \mathbf{x} , as well as the objective function that is optimized over the inter-appointment times $\min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma, \mathbf{x})$. In Subsection 4.1 we give a closed-form expression for the objective function exploiting the structure of phase-type distributions. Then, in Subsection 4.2, we propose an approximation of the objective function based on a heavy-traffic regime, leading to an expression that allows an explicit minimization over $\mathbf{x} \in \mathbb{R}_+^n$. A hybrid version that can efficiently compute good appointment schedules is introduced in Subsection 4.3.

Throughout this section, we assume that the tour σ is given. To avoid excessive notation, we let j here denote the index of the j -th visited client in the given σ (that is, we use j instead of σ_j). Furthermore, the analysis here focuses on the idle and waiting times, i.e., the second and third terms of the objective function in (1).

4.1. Phase-type method to evaluate objective function. As alluded to in Section 3, the evaluation of $\mathcal{L}(\sigma, \mathbf{x})$ is challenging, even for a given tour σ and given inter-appointment times \mathbf{x} . This can be observed from the Lindley recursion; the evaluation of $\mathbb{E}W_j$ requires that we know the distribution of W_{j-1} . One cannot obtain closed-form expressions for the distribution function of the idle and waiting times for general travel and/or service times. This motivates why we follow a phase-type method for approximating $\mathbb{E}I_j$ and $\mathbb{E}W_j$ for any travel and service time distribution, which has been advocated, and firmly backed, in [25]. Before presenting the two-step method, let us first give some relevant preliminaries and motivation related to phase-type distributions.

We first recall that any distribution on the positive half line can be approximated arbitrarily closely by a *phase-type distribution*; we formally say that the class of phase-type distributions is dense in the class of all distributions on $(0, \infty)$ [3, Thm. III.4.2]. The class of phase-type distributions contains mixtures and convolutions of exponential distributions, and thus includes (mixtures of) Erlang distributions and hyperexponential distributions; formally a phase-type random variable is defined as the entrance time of an absorbing state in a continuous-time Markov chain [3, Section III.4]. In particular, a phase-type distribution is characterized through the triple (d, α, V) with d the number of transient states of the Markov chain, α an initial distribution, and V the transition rate matrix. We refer to Appendix A for some background on phase-type distributions; in particular, we provide the specific definitions of the ones that play a role in this paper.

In [38] it is pointed out how the distributions of I_j and W_j can be (numerically) evaluated in case $U_j = T_j + B_{j-1}$ is of phase-type; indeed, applying techniques from linear algebra, a recursive technique to determine the cumulative distribution functions $\mathbb{P}(I_j \leq t)$ and $\mathbb{P}(W_j \leq t)$, for $t \geq 0$, has been devised. We adopt a slightly different approach to evaluate the objective function $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$.

Now, our phase-type method consists of the following two steps: (1) approximate the sum of travel and service times for each client by a phase-type distribution, and (2) provide an explicit formula for the objective function in case of phase-type service and travel times. For step (1), the specific phase-type distributions that we will be using are mixtures of Erlang distributions and the hyperexponential distribution, see [34] for additional justification. The underlying idea is to fit the parameters of these distributions such that they match the first two moments of the distribution of the target random variable, say U (or, equivalently, the first moment $\mathbb{E}U$ and the *squared coefficient of variation* $\text{SCV}(U) := \text{Var } U / (\mathbb{E}U)^2$). Fitting the first two moments means that evidently some detail about the distribution of U is lost. It should, however, be borne in mind that (i) in general only lower moments of the random variables under consideration can be reliably estimated from historical data, and (ii), as backed by the experiments in [21], the approximation of random variables by their phase-type counterparts is highly accurate. In the sequel we denote by \bar{U} the phase-type random variable that approximates U .

- In the case of \bar{U} being a mixture of Erlang distributions, \bar{U} corresponds to an Erlang distributed random variable with $K - 1$ phases and mean $(K - 1)/\mu$ with probability p , and with an Erlang distributed random variable with K phases and mean K/μ with probability $1 - p$; we write that \bar{U} is distributed as $E_K(\mu, p)$. As pointed out in [34], for such a \bar{U} , with $K \in \{2, 3, \dots\}$, the squared coefficient of variation (SCV) satisfies:

$$\text{SCV}(\bar{U}) = \frac{\text{Var } \bar{U}}{(\mathbb{E} \bar{U})^2} = \frac{K - p^2}{(K - p)^2} \in (K^{-1}, (K - 1)^{-1}].$$

As a consequence, if $\text{SCV}(U) \in (K^{-1}, (K - 1)^{-1}]$, then we propose to approximate U by $E_K(\mu, p)$ with

$$p = \frac{1}{1 + \text{SCV}(U)} \left(\text{SCV}(U)K - \sqrt{(1 + \text{SCV}(U))K - \text{SCV}(U)K^2} \right) \quad \text{and} \quad \mu = \frac{K - p}{\mathbb{E}U},$$

such that U and \bar{U} have the same first two moments, cf. [34].

- In the case of \bar{U} being hyperexponential, \bar{U} corresponds with probability p to an exponential distribution with rate μ_1 , and with probability $1 - p$ to an exponential distribution with rate μ_2 ; we write that \bar{U} is distributed as $H_2(\mu_1, \mu_2, p)$. When fitting the first two moments with three parameters, there is clearly a degree of freedom. To remedy this, we choose $\mu_1 = 2p\mu$ and $\mu_2 = 2(1 - p)\mu$ for some $\mu > 0$ (a choice referred to as ‘balanced means’). An elementary computation reveals that

$$\text{SCV}(\bar{U}) = \frac{1}{2p(1 - p)},$$

which is larger than or equal to 1. Hence, if $\text{SCV}(U) \geq 1$ it can be verified (see [34]) that one can pick

$$p = \frac{1}{2} \left(1 + \sqrt{\frac{\text{SCV}(U) - 1}{\text{SCV}(U) + 1}} \right), \quad \mu_1 = \frac{2p}{\mathbb{E}U}, \quad \mu_2 = \frac{2(1 - p)}{\mathbb{E}U},$$

to match the first two moments of U and \bar{U} .

Turning to step (2), we present an approach in Appendix B, based on [38], to evaluate the objective function $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$ using that all \bar{U}_j are of phase type. More specifically, we identify the distribution of the random variables $R_j := W_j + \bar{U}_{j+1}$, which is in fact of phase type itself. As shown in the appendix, the

phase-type distribution of R_j is characterized by the triplet $(D_j, \alpha_{(j)}, V^{(j)})$; the parameters are recursively defined and can be found in Appendix B. An explicit formula for the objective function is given in the following theorem, whereas the derivation is deferred to Appendix B. Observe that the expression for the objective function is exact in case all U_j are phase type; if not all U_j are phase-type, the theorem provides an accurate approximation.

Theorem 1. *If all U_j are phase type, then, for $\sigma \in \mathcal{J}, \mathbf{x} \in \mathbb{R}_+^n$,*

$$\mathcal{L}(\sigma, \mathbf{x}) = \omega^T \sum_{j=1}^{n+1} \mathbb{E} T_j + \sum_{j=1}^n \left(\omega^I \left(x_j + \alpha_{(j)} (V^{(j)})^{-1} \mathbf{1} - \alpha_{(j)} (V^{(j)})^{-1} e^{V^{(j)} x_j} \mathbf{1} \right) + \omega_j^W \left(-\alpha_{(j)} (V^{(j)})^{-1} e^{V^{(j)} x_j} \mathbf{1} \right) \right),$$

with $\alpha_{(j)}$ and $V^{(j)}$ recursively defined in Appendix B and $\mathbf{1}$ a vector of ones.

When it comes to our RAS problem, the above methodology has one serious limitation. The expressions in Appendix B reveal that, in particular when some of the random variables U_j have a low SCV, the dimension of the matrices can become prohibitively large. In addition, evaluating the objective function requires a number of matrix inverses and matrix exponentials, which are computationally expensive. The optimal appointment times can be numerically obtained, e.g., using gradient descent methods, by exploiting the fact that the objective function is convex in \mathbf{x} , see [25]. Although a single evaluation does not take much time in practice, obtaining the optimal solution

$$\mathcal{L}_{\text{ph}}(\sigma) := \min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma, \mathbf{x}),$$

requires many iterations, for which this approach becomes infeasible. Instead of optimizing over the inter-appointment times, we present a heavy-traffic approximation in the next section.

4.2. Heavy-traffic approximation. In this subsection we discuss a heavy-traffic approximation of our objective function, which allows us to easily optimize over the inter-appointment times. It is expected to be particularly accurate when one cares more about idle times than about waiting times. The reason is that in that regime the schedule will be such that the queueing system, as defined through the Lindley recursion, will be operating under heavy load.

The first step is that, based on the heavy-traffic approximation for the mean waiting time in a GI/G/1 queue [3, Section X.7], we propose the approximation

$$\mathbb{E} W_j \approx \frac{S_j}{2(x_j - \mathbb{E} U_j)},$$

where we denote

$$S_j := \frac{\sum_{i=1}^j \beta^{j-i} \text{Var } U_i}{\sum_{i=1}^j \beta^{j-i}},$$

where empty sums being defined to equal 0. The constant $\beta \in [0, 1]$ represents the rate of the exponential decay in the variance of the service and travel time that a client has on subsequent clients. After some numerical experiments, we chose $\beta = 0.5$. As this approximation is based on results for a stationary random variable, we assume that $x_j > \mathbb{E} U_j$. In addition, for the mean idle times, we can use the intuitive approximation

$$\mathbb{E} I_j \approx x_j - \mathbb{E} U_j.$$

Upon combining the above approximations, we obtain the following heavy-traffic approximation of the objective function:

$$\mathcal{L}_{\text{ht}}(\boldsymbol{\sigma}, \mathbf{x}) := \omega^{\text{T}} \sum_{j=1}^{n+1} \mathbb{E} T_j + \omega^{\text{I}} \sum_{j=1}^n (x_j - \mathbb{E} U_j) + \sum_{j=1}^n \omega_j^{\text{W}} \frac{S_j}{2(x_j - \mathbb{E} U_j)}.$$

We can thus try to find an approximate value of \mathcal{L}^* by performing the optimization problem

$$\mathcal{L}_{\text{ht}}^* := \min_{\boldsymbol{\sigma} \in \mathcal{J}, \mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}_{\text{ht}}(\boldsymbol{\sigma}, \mathbf{x}). \quad (\text{RAS-ht})$$

Interestingly, it is possible to *explicitly* solve the optimization over \mathbf{x} . To this end, first note that it is straightforward to verify that this function is convex in its arguments x_1, \dots, x_n . By solving its first-order conditions, we have that for $j = 1, \dots, n$,

$$x_j^{(\text{ht})} = \mathbb{E} U_j + \sqrt{\frac{\omega_j^{\text{W}} S_j}{2\omega^{\text{I}}}}. \quad (2)$$

These inter-appointment times are intuitively appealing and turn out to provide good appointment schedules. Note that these findings are consistent with those for the stationary homogeneous case under heavy traffic, as have been presented in [24]. As an aside, we mention that the corresponding value of the objective function can now be evaluated in closed form.

Proposition 1. *For all $\boldsymbol{\sigma} \in \mathcal{J}$,*

$$\mathcal{L}_{\text{ht}}(\boldsymbol{\sigma}) := \min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}_{\text{ht}}(\boldsymbol{\sigma}, \mathbf{x}) = \mathcal{L}_{\text{ht}}(\boldsymbol{\sigma}, \mathbf{x}^{(\text{ht})}) = \omega^{\text{T}} \sum_{j=1}^{n+1} \mathbb{E} T_j + \sqrt{2\omega^{\text{I}}} \sum_{j=1}^n \sqrt{\omega_j^{\text{W}} S_j}, \quad (3)$$

with $\mathbf{x}^{(\text{ht})}$ given by (2).

Remark 1. Importantly, for the case without travel times (e.g., the conventional appointment scheduling situation where $\mathbb{E} T_j = 0$) the expression (3) for the approximate minimum cost offers additional support for the SVF rule: as can be seen directly, it is minimized if the clients are sequenced in increasing order of variance.

4.3. Hybrid approximation. In Subsection 4.1, we presented a phase-type method to obtain an accurate approximation of the objective function $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$; in fact, this method is exact when travel and service times follow a phase-type distribution. Although this method is accurate, it is relatively slow, in particular if we aim to find optimized inter-appointment times \mathbf{x} . In Subsection 4.2, we derived a heavy-traffic approximation, which is less accurate than the phase-type method, but it is fast. Based on numerical experiments, presented in Appendix D, we propose a hybrid version that combines the strengths of the two approaches:

$$\mathcal{L}_{\text{hyb}}(\boldsymbol{\sigma}) := \mathcal{L}(\boldsymbol{\sigma}, \mathbf{x}^{(\text{ht})}), \quad (4)$$

with $\mathbf{x}^{(\text{ht})}$ the vector whose entries are given by (2). This hybrid version can be seen as ‘best of both worlds’: the appointment times stem from the heavy-traffic approximation, i.e., require virtually no computation time. These appointment times are then inserted into the phase-type based objective function presented in Theorem 1 and yields a more accurate proxy of the objective function than when $\mathbf{x}^{(\text{ht})}$ is inserted into the heavy-traffic based objective function, i.e., Proposition 1.

5. LARGE NEIGHBORHOOD SEARCH METAHEURISTIC

The previous section presented an approximation to the RAS problem, transforming it to a pure routing problem. In this section, we propose the use of a large neighborhood search (LNS) metaheuristic to efficiently explore the search space of routing solutions. LNS was first introduced by [30]; a recent and more general treatment is given by [27], including many successful examples of the LNS metaheuristic in solving combinatorial optimization problems. The main idea underlying LNS is to improve an initial solution (i.e., a tour) by iteratively *destroying* and *repairing* the solution.

5.1. Algorithm outline. A general outline of the LNS metaheuristic is given in Algorithm 1. The algorithm takes an initial solution and initializes it as the best solution. In each iteration of the algorithm, a random pair of destroy $d(\cdot)$ and repair $r(\cdot)$ operators is selected and applied to the current tour, resulting in a candidate solution σ^c . An acceptance criterion determines whether this candidate solution is accepted as the solution for the next iteration or rejected in favor of the old solution σ . Moreover, if the identified candidate solution improves over the best-found solution so far, we register it as the new best solution σ^b . We iterate this procedure until a provided time limit is met, after which we return the best-found solution.

Algorithm 1: Large neighborhood search

Input: Initial solution σ

Output: The best-found solution σ^b

```

1  $\sigma^b \leftarrow \sigma$ 
2 while time limit is not exceeded do
3     Select destroy and repair operator pair  $(d, r)$  at random
4      $\sigma^c \leftarrow r(d(\sigma))$ 
5     if  $\sigma^c$  is accepted then
6          $\sigma \leftarrow \sigma^c$ 
7     if  $\sigma^c$  has a better objective value than  $\sigma^b$  then
8          $\sigma^b \leftarrow \sigma^c$ 
9 return  $\sigma^b$ 
    
```

5.2. Details of the algorithm. This section describes the details of the LNS algorithm in the context of the RAS problem.

5.2.1. Objective function. A candidate solution σ^c is evaluated using the objective function $\mathcal{L}_{\text{hyb}}(\sigma^c)$ defined in (4).

5.2.2. Initial solution. We use a random permutation of client visits as the initial solution.

5.2.3. Destroy and repair operators. We consider two destroy operators. The first one is the *random client* operator, which randomly removes between 1 and D clients from the current solution, with $D \in \mathbb{N}$ a parameter of the algorithm. The second one is an *adjacent client* operator, which removes a randomly selected sequence of between 1 and D adjacent clients. For the repair operator, we consider a *greedy insert* repair operator. This operator inserts each of the removed clients into the position that increases the objective function by the least amount, one by one.

5.2.4. *Acceptance criterion.* We use the record-to-record-travel acceptance criterion [9]. This criterion accepts a new candidate solution σ^c if $\mathcal{L}_{\text{hyb}}(\sigma^c) - \mathcal{L}_{\text{hyb}}(\sigma^b)$ is smaller than some updating threshold H for the current best-found solution σ^b . This threshold is initialized at some starting value $H_0 \geq 0$ and is equal to $H = H_0 \cdot t/t_{\max}$, where t denotes the elapsed runtime and $t_{\max} \geq t \geq 0$ denotes the maximum time limit.

5.2.5. *Post-termination optimization.* After termination of the LNS algorithm, we use the best-found tour σ^b to compute the optimal appointment schedule \mathbf{x}^* , i.e.,

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma^b, \mathbf{x}). \quad (5)$$

The final solution to the RAS problem is then given by (σ^b, \mathbf{x}^*) . In summary, the idea is that to determine the tour, we solely use the hybrid approximations with appointment schedules based on heavy traffic; for the best-found tour, we finally determine the appointment schedule based on the phase-type method of the objective function.

6. NUMERICAL EXPERIMENTS

In this section, we present the numerical experiments. We describe the generation procedure of benchmark instances in Subsection 6.1, the implemented benchmarks algorithms in Subsection 6.2, parameter tuning in Subsection 6.3 and the results in Subsection 6.4. All instances, algorithms, and results are openly available at <https://github.com/leonlan/routing-appointment-scheduling>.

6.1. **Instance generation procedure.** We followed the instance generation procedure described in [43], which is common in the integrated routing and appointment scheduling literature. An instance is defined on a $[0, 50]^2$ square grid in which the depot is located at coordinate $(0, 0)$ and n clients are uniformly randomly dispersed. The mean travel time between two locations equals the Euclidean distance, and we consider a fixed SCV for travel times of 0.15. The mean service times are sampled from $U(30, 60)$, where $U(a, b)$ denotes the uniform distribution on $[a, b]$. The SCVs for service times are either all sampled from $U(0.15, 0.5)$ or from $U(0.5, 1.5)$. We refer to these two cases as *low* and *high* service time SCVs. The weights for client waiting times ω_j^W are sampled from $U(1, 10)$, and the weight of the operator's idle time ω^I is fixed at 2.5. To investigate the trade-off between travel times and appointment costs, we consider three scenarios with ω^T being 0.5, 1, and 2, respectively.

For a given instance size n , we sampled 20 different realizations for all parameters except for the SCVs of service times and the weights for travel times. Using these base instances, we sampled the SCVs of service times for the low and high variance scenarios and used each of the three different ω^T values. This resulted in $20 \times 2 \times 3 = 120$ unique problem instances for a given instance size n .

6.2. **Benchmark algorithms.** We implemented three benchmark heuristics to compare against our LNS metaheuristic. All three heuristics first identify a tour of the client visits, after which the optimal appointment schedule is computed, similar to the procedure outlined in Subsection 5.2.5.

- The first heuristic is the modified TSP (MTSP) heuristic proposed by [43]. The MTSP solves a deterministic TSP in which the edge weights are modified to take into account appointment scheduling costs. A more detailed explanation of the algorithm is given in Appendix B.
- The second heuristic solves the deterministic TSP using the mean travel times, yielding the tour $\sigma^{\text{TSP}} = (\sigma_1^{\text{TSP}}, \dots, \sigma_n^{\text{TSP}})$. For this tour, the objective value follows by optimizing over the appointment times for tour σ^{TSP} , i.e., $\min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma^{\text{TSP}}, \mathbf{x})$. Since the benchmark instances assume Euclidean distances, the reverse direction of the tour $(\sigma_n^{\text{TSP}}, \dots, \sigma_1^{\text{TSP}})$ provides the same travel time. Hence, we take the direction of the tour with the smallest objective value of the two

directions. From our numerical experiments, we have observed that a considerable gain can be achieved by considering the two-sided tour compared to the one-sided version.

- The third heuristic is a modified version of the smallest variance first rule (MSVF) that incorporates stochastic travel times. Starting at the depot, we select the next, not-yet-visited client with the lowest variance of combined travel and service time. That is, given a partial tour $\sigma' = (\sigma_1, \dots, \sigma_k)$, we select the next location by

$$\arg \min_{j \in V, j \notin \sigma'} \text{Var}(T_{\sigma_k, j} + B_j).$$

We repeat this procedure until all clients have been visited. For the identified tour, we compute the optimal inter-appointment times similar to the TSP algorithm as described above. When the variance of the travel times goes to zero, MSVF reduces to the classical SVF rule where clients are visited in order of smallest variance of the service times.

6.3. Parameter tuning. For LNS, we performed a full factorial design using the parameters D (maximum number of removed clients) and H_0 (starting threshold of acceptance criterion). We restricted the values to $D \in \{2, 3, \dots, 12\}$ and $H^{\text{init}} \in \{0.01, 0.025, 0.05, 0.075, 0.10\}$, where we set H_0 equal to H^{init} multiplied with the objective value of the initial solution. This resulted in 55 different parameter combinations. All combinations were evaluated on a set of 60 training instances of size $n \in \{10, 15, 20\}$ generated following the same procedure as the benchmark instances. The time limits were set to 15, 30 and 60 seconds for $n = 10, 15$ and 20, respectively. The best results were obtained for $D = 6$ and $H^{\text{init}} = 0.05$, hence we selected these parameters.

6.4. Results. We now present the results of the numerical experiments. Subsection 6.4.1 presents the results for small benchmark instances with $n \in \{6, 8, 10\}$ and Subsection 6.4.2 presents those for large instances with $n \in \{15, 20, 25, 30, 35, 40\}$. Subsection 6.4.3 investigates the runtime performance of the LNS algorithm and of computing the optimal appointment schedules. Finally, Subsection 6.4.4 performs a sensitivity analysis of the algorithmic performance when shifting the emphasis from the routing to the appointment scheduling component by scaling the travel times.

We implemented all our numerical experiments in Python 3.11. We used the LKH-3 solver [13] to obtain solutions to the deterministic TSP, and we implemented the LNS metaheuristic using the ALNS Python package [40]. The optimal appointment schedules were obtained using trust-region methods from the SciPy Python package [37]. Each instance was solved on a single core of an AMD EPYC 7H12 CPU.

For each solved instance, we computed the *gap*, which is defined as the relative difference in objective value between the obtained solution and the best-found solution. The tables report the average gap over all instances for specific instance categories.

6.4.1. Small instances. The first set of numerical experiments compares the performance of LNS and the benchmark algorithms against the (near-)optimal solution on small problem instances with $n \in \{6, 8, 10\}$. We can obtain the optimal solution by fully enumerating all possible tours and computing the corresponding optimal appointment schedules. For instances with $n = 6$ and $n = 8$, the optimal solution can be computed within reasonable time limits (below, say, 1 hour). However, for instances with $n = 10$ clients, the time needed to evaluate a single instance would take roughly 500 hours on a single core; hence, for this value of n we resorted to a variant of the full enumeration approach. More, specifically, we first enumerated all $10!$ tours and evaluated them using the heavy traffic appointment schedule $\mathbf{x}^{(\text{ht})}$, giving $\mathcal{L}_{\text{ht}}(\sigma)$ for all $\sigma \in \mathcal{S}$. We then considered the $8!$ solutions with the lowest objective value and evaluated the tours from these solutions again using the optimal appointment schedule, yielding $\mathcal{L}_{\text{ph}}(\sigma)$ for the $8!$

n	ω^T	Low SCV				High SCV			
		LNS	MTSP	TSP	MSVF	LNS	MTSP	TSP	MSVF
6	0.5	0.80	5.70	4.50	11.16	2.59	10.96	6.83	11.00
	1.0	0.73	4.82	3.43	13.16	1.96	8.15	5.65	12.70
	2.0	0.79	2.71	2.45	16.50	1.97	7.23	4.20	15.75
8	0.5	1.12	5.25	3.73	9.46	3.93	10.35	7.28	9.69
	1.0	0.84	4.47	2.88	12.17	2.84	7.90	5.68	11.39
	2.0	0.56	3.99	1.96	16.43	2.24	6.36	3.96	14.96
10	0.5	2.04	7.72	5.50	10.27	3.32	10.47	8.64	11.16
	1.0	1.14	5.86	4.10	13.71	3.00	8.55	6.82	13.64
	2.0	0.88	4.63	2.80	19.74	1.70	5.82	4.62	18.56
Avg.		0.99	5.02	3.48	13.62	2.62	8.42	5.96	13.21

TABLE 1. Results on small benchmark instances. Values represent the average gaps per instance size, travel time weight and SCV variation. The best average gap per category is marked in bold.

considered tours σ . Although this does not guarantee an optimal solution, we believe that the obtained solutions are near-optimal and serve as reliable reference solutions.

The time limit for LNS was set to 5, 10, and 15 seconds for $n = 6, 8$, and 10, respectively. The runtimes of MTSP, TSP, and MSVF were negligible. Table 1 shows the results for different instance sizes n , travel time weights ω^T , and SCV variation. On average, LNS achieved the lowest average gap of 1.80% on all small instances (0.99% for low SCV and 2.62% for high SCV), whereas the average gaps of MTSP, TSP, and MSVF were 6.72%, 4.72%, and 13.41%, respectively. The results reveal that the integration of routing and appointment scheduling decisions in our LNS metaheuristic has clear benefits over simpler heuristics. TSP performed surprisingly well: in our results, the TSP algorithm outperformed the MTSP and MSVF heuristics. We noticed in our experiments that evaluating both orientations for the TSP algorithm was a prominent reason for this: evaluating only one orientation of the TSP tour resulted in considerably worse solutions. When evaluating only one orientation of the TSP tour, the results of MTSP, TSP, and MSVF were consistent with those presented in [43]. The performance of MSVF is rather poor across all instances, as the emphasis on travel times in the instances is relatively large.

When comparing results between the different values of the travel time weight, we observe that LNS obtains lower gaps for high values, that is, when there is more emphasis on routing. Similarly, the results show that for low service time SCVs, the average gap achieved by LNS is lower than in the case of high service time SCVs (0.99% versus 2.62%). This clearly indicates that problems that have more emphasis on the appointment scheduling problem, i.e., low travel time weights and higher variance in service times, result in generally more challenging instances, clearly showing that there is still room for improvement.

6.4.2. Large instances. In the second set of experiments, we compare the performance of LNS and the benchmark algorithms on larger problem sizes with $n \in \{15, 20, 25, 30, 35, 40\}$. The time limit for LNS was set to 30, 60, 120, 240, 480, and 960 seconds, respectively. Note that it is no longer feasible to compute

the (near-)optimal solution using full enumeration, hence the presented gaps are with respect to the best-found solution.

n	ω^T	Low SCV				High SCV			
		LNS	MTSP	TSP	MSVF	LNS	MTSP	TSP	MSVF
15	0.5	0.16	3.75	2.85	8.95	0.16	5.76	4.10	6.41
	1.0	0.16	2.45	2.03	13.59	0.09	4.04	3.26	10.13
	2.0	0.70	1.88	1.70	22.15	0.18	2.79	2.19	17.01
20	0.5	0.28	3.41	2.29	8.53	0.03	5.99	5.20	7.21
	1.0	0.53	2.85	1.88	13.83	0.20	3.74	3.76	10.57
	2.0	1.44	1.69	1.36	22.91	0.21	2.76	2.78	18.15
25	0.5	0.30	2.13	1.67	7.98	0.00	5.96	5.41	7.26
	1.0	1.01	1.00	1.07	13.12	0.03	4.80	4.61	11.34
	2.0	1.34	1.22	0.58	22.48	0.10	3.33	2.98	18.43
30	0.5	0.21	3.11	1.64	7.11	0.00	5.82	5.40	5.98
	1.0	0.70	2.09	0.84	12.22	0.00	5.55	4.42	9.68
	2.0	3.49	1.05	0.27	22.01	0.01	3.67	2.47	16.15
35	0.5	0.61	1.81	0.86	6.64	0.01	6.19	5.26	6.57
	1.0	1.95	1.39	0.47	11.92	0.11	4.87	4.14	10.21
	2.0	4.97	1.25	0.17	21.68	0.34	3.65	2.85	17.66
40	0.5	0.57	1.80	0.46	5.99	0.00	7.19	6.10	6.68
	1.0	1.77	1.46	0.11	11.31	0.00	6.18	4.98	10.26
	2.0	5.85	1.04	0.06	21.48	0.00	4.15	2.81	16.65
Avg.		1.45	1.97	1.13	14.11	0.08	4.80	4.04	11.46

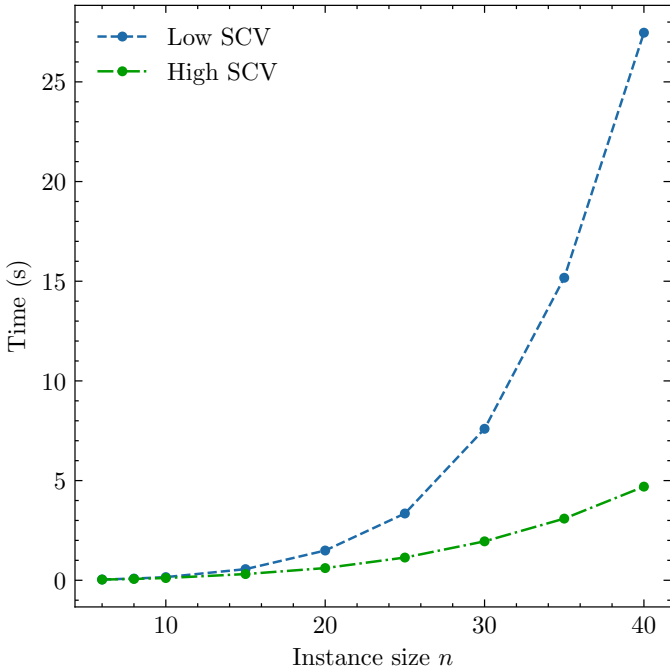
TABLE 2. Results on large benchmark instances. Values represent the average gaps per instance size, travel time weight and SCV variation. The best average gap per category is marked in bold.

Table 2 shows results for larger instances. LNS achieved an average gap on all instances of 0.77%, whereas MTSP, TSP and MSVF achieved an average gap of 3.38%, 2.58%, 12.78%, respectively. Although the average gap between LNS and other algorithms has decreased compared to the results on small instances, the results still show that LNS outperformed the other heuristics on average. Specifically, for instances with high SCVs, LNS consistently outperforms the other heuristics, with an average gap difference of 3.96% when compared to TSP. However, LNS no longer obtains the best results for instances with low SCVs: it is generally outperformed by TSP when instances become larger and the travel time weights are higher. This is a result of the high computational complexity of evaluating the objective function: phase-type distributions with low SCVs require more phases, and thus yield larger matrices.

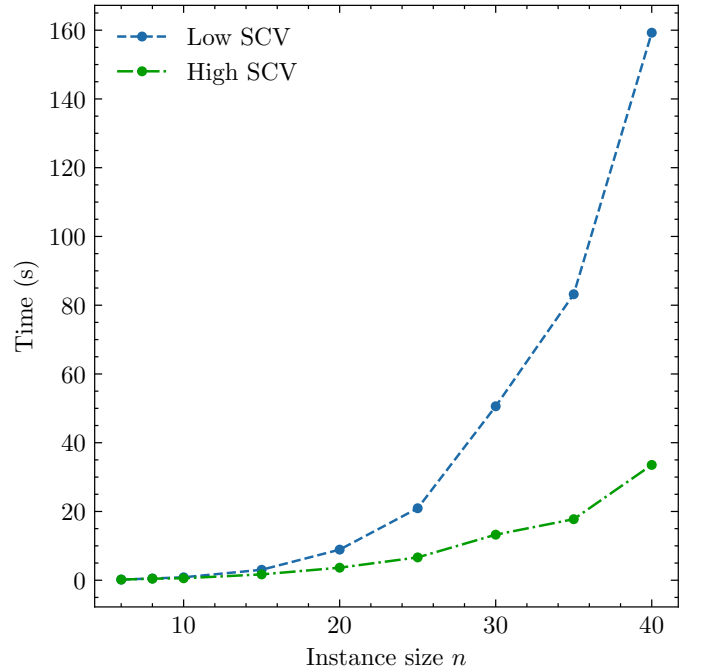
We note that the performance of LNS can be improved by increasing the time limit. The impact of the runtime is discussed in more detail in the next section.

6.4.3. Analysis of the runtimes. We provide here more details about the runtimes of the algorithms. First, we show how much time one single LNS iteration needs on average. And, second, we show how much time is required to compute the optimal appointment schedule when the final tour has been selected. Both evaluations rely on evaluating the objective function presented in Theorem 1. The time complexity to evaluate this function grows fast: it requires general matrix operations such as multiplication and inversion, as well as the computation of the matrix exponential. One important factor that determines the size of the matrices is the SCVs of the combined service and travel times.

Figure 3 visualizes the average runtimes for both a single LNS iteration $\mathcal{L}_{\text{hyb}}(\sigma)$ and computing the optimal appointment times $\mathcal{L}_{\text{ph}}(\sigma)$ for some tour σ . The runtime values were obtained by solving the benchmarking instances in the previous sections. Figure 3a shows the average runtime for a single LNS iteration. For the low SCV case, the plot shows that a single LNS iteration can take up to 30 seconds for $n = 40$ clients. In view of the time limit of 960 seconds that we used for instances with $n = 40$, this means that LNS was only able to run roughly thirty iterations to solve the instance. On the other hand, for high SCVs, the average runtime is 5 seconds for $n = 40$, meaning that it would have up to 200 iterations. Figure 3b shows the average runtime for computing the optimal appointment schedule after any algorithm has decided on the final tours. Note that this time is not included in the LNS time limit. For high SCVs, computing the optimal schedule takes up to 40 seconds with $n = 40$ clients, whereas for low SCVs it can take up to 160 seconds. Both figures indicate that for cases with low variance, there is a very rapid increase in computational burden which renders a local search-based approach like LNS infeasible for large instances. For cases with higher variance, the performance is considerably better, and it can consistently outperform the considered heuristics.



(A) LNS iteration



(B) Optimal schedule

FIGURE 3. Average runtimes in seconds (A) for one LNS iteration and (B) to compute optimal appointment times.

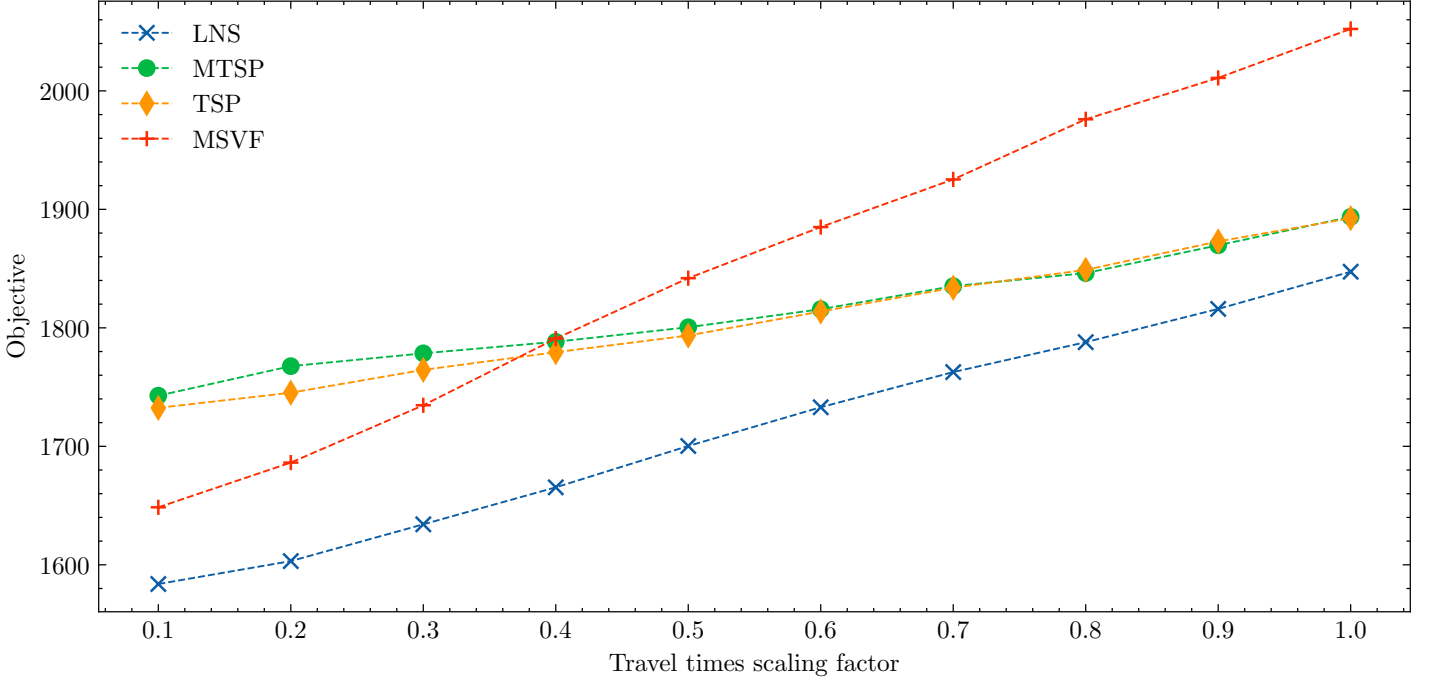


FIGURE 4. Influence of the travel time scaling factor on the average objective for each algorithm.

6.4.4. Sensitivity analysis. In Subsections 6.4.1 and 6.4.2, we showed that MSVF is outperformed by the other algorithms, which indicates that the benchmark instances emphasize good routing decisions over appointment scheduling decisions. We note that the impact of routing is amplified by the fact that a relatively short tour coincides with a smaller variance in travel times, in turn leading to smaller idle and waiting times. Hence, a good routing strategy provides additional benefits for appointment scheduling decisions.

In the final set of experiments, we performed a sensitivity analysis by shifting the emphasis from routing to appointment scheduling by scaling the travel times. To this end, we used the set of benchmark instances with $n = 15$ with both low and high SCV and a fixed travel time weight ω^T of 1. To emphasize the relative importance of appointment scheduling, we scaled the travel times with a scaling factor, for which we used values from 0.1 to 1 with increments of 0.1. The service times were unmodified.

Figure 4 shows the average objective value obtained by each algorithm as a function of the scaling factor. As the travel-time scaling factor decreases, the relative performance of MSVF compared to the TSP-based algorithms improves, as there is more emphasis on the appointment scheduling aspect of the RAS problem. For a scaling factor of 0.4, the MTSP, TSP, and MSVF heuristics perform about equally well; MSVF outperforms the TSP-based algorithms in case the scaling factor is smaller than 0.4. LNS demonstrates superior performance compared to the heuristics across all instances. In particular, when the importance of routing and appointment scheduling are well balanced (with a scaling factor between 0.3 and 0.5) the gap between LNS and the best of the other algorithms is considerable. Overall, in situations with a weighted objective of routing and appointment scheduling, LNS is able to efficiently take both aspects into account, demonstrating the benefit of integrated decision-making.

7. DISCUSSION AND CONCLUDING REMARKS

In this paper, we have studied the integrated routing and appointment scheduling (RAS) problem, with the complicating feature that service and travel times are assumed to be of a stochastic nature. Our

approach phrases the problem in terms of queueing-theoretic concepts, which is in contrast with the more commonly applied scenario-based approaches. First, we point out how the problem’s objective function can be evaluated when all random variables involved are of phase-type; this is attractive because one can accurately approximate any random variables phase-type counterparts, but the downside is that the evaluation of an optimal appointment schedule can be slow, in particular for larger numbers of clients. This has motivated the exploration of a second class of approximations, based on the celebrated heavy-traffic approximation from queueing theory; while sacrificing some accuracy, the closed-form formulas allow extremely fast evaluation of the objective function. A hybrid method combines the strong aspects of both approaches: by the heavy-traffic approximation the optimal appointment times are evaluated, which are then inserted in the phase-type objective function. The use of this hybrid method effectively reduces the RAS problem to a more conventional routing problem, that we propose to solve using large neighborhood search.

The effectiveness of our approach is demonstrated through an extensive set of computational experiments, with up to 40 clients. These experiment show that it outperforms other heuristics in (almost) all instances. Overall, the gap between our approach and TSP-based heuristics tends to become smaller when the travel time is relatively large compared to the service time, the weight of the travel time is relatively large, and the variability in service times is small.

In instances that the coefficient of variation of the service times is small and the number of clients is large (say, in the order of 40), the computation time of our approach starts to explode. This is due to the fact that when using the hybrid method, the phase-type objective function still needs to be evaluated once per every iteration, which becomes prohibitively slow due to the matrix operations required (matrix exponentials, inverse matrices, etc.). If we wish to scale up our method so as to be able to accommodate larger numbers of clients, we need to find an alternative to the currently used phase-type evaluation of the objective function.

We envisage great potential of our queueing-based approach in similar types of optimization problems. One could think of the situation of multiple service operators, where in addition to the routing and appointment decisions, one needs to determine which clients are to be served by each of the servers; one could add even one more layer in which and the number of service operators has to be chosen. Moreover, one could incorporate randomness in the number of clients, e.g., due to last-minute cancellations or unscheduled ‘high-priority’ clients. Other possible extensions could relate to taking into account client preferences into the appointment times.

APPENDIX A. PHASE-TYPE DISTRIBUTIONS

Consider a discrete-time Markov chain $\{J_t\}_{t \geq 0}$ that is defined on the finite state space $\{1, \dots, d+1\}$. Here, all states $1, \dots, d$ are transient and state $d+1$ is absorbing. Then for $\{J_t\}_{t \geq 0}$ having an initial distribution $\boldsymbol{\alpha} \in \mathbb{R}^d$ (which is to be understood as a *row* vector), a phase-type random variable is defined as the time it takes to reach state $d+1$. We then define a transition rate matrix $V \equiv (v_{kl})_{k,l=1}^d$ as follows. Denote by v_{kl} , with $k \neq l$ and $k, l = 1, \dots, d$, the transition rate from transient state k to transient state l ; the rate from state k to the absorbing state is denoted by v_k , for $k = 1, \dots, d$. The diagonal elements of the matrix V are chosen such that $\mathbf{v} = -V\mathbf{1}$, where $\mathbf{1}$ is a vector of all ones of appropriate dimension. A phase-type distribution is characterized through the triple $(d, \boldsymbol{\alpha}, V)$.

For such phase-type distributions, a large set of results are known; see, for instance, [3, Section III.4]. In the context of the present paper, the most relevant result concerns the corresponding probability density function: it is given by $f(y) = \boldsymbol{\alpha} e^{Vy} \mathbf{v}$.

Two special phase-type distributions play a crucial role in this paper: the mixture of Erlang distributions $E_K(\mu, p)$ and the hyperexponential distribution $H_2(\mu_1, \mu_2, p)$.

- For the mixture of Erlang distributions $E_K(\mu, p)$ we have $d = K$, $\boldsymbol{\alpha} = (1-p, p, 0, \dots, 0)$ and

$$V = \begin{pmatrix} -\mu & \mu & 0 & \cdots & 0 \\ 0 & -\mu & \mu & \cdots & 0 \\ 0 & 0 & -\mu & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\mu \end{pmatrix}.$$

- For the hyperexponential distribution $H_2(\mu_1, \mu_2, p)$ we have $d = 2$, $\boldsymbol{\alpha} = (p, 1-p)$ and

$$V = \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}.$$

APPENDIX B. PHASE-TYPE METHOD FOR THE OBJECTIVE FUNCTION

In this appendix we point out how the objective function $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$ can be numerically evaluated given we know the phase-type description \bar{U}_j of the random variables U_j ; that is, we present the derivation of Theorem 1. The procedure uses various concepts and ideas developed in [38]. As in Subsection 4.1, we assume that the tour $\boldsymbol{\sigma}$ is given, where j denotes the index of the j -th visited client in the given tour $\boldsymbol{\sigma}$.

The following lemma plays a crucial role. Define by R_j the *sojourn time* of the j -th client, which is in this context defined as the time elapsed between the appointment time and the moment that the service of the $(j+1)$ -st can start; i.e., $R_j = W_j + U_{j+1}$. Note, importantly, that this means that in the present setting in which the server travels between locations, the sojourn time of the j -th client is *not* defined as the time she spends in the system (i.e., $W_i + B_i$). For $j = 1, \dots, n$, we define

$$\begin{aligned} \mathcal{R}_j(\mathbf{x}) &:= \omega^I \mathbb{E}(x_j - R_{j-1}) 1\{x_j \geq R_{j-1}\} + \omega_j^W \mathbb{E}(R_{j-1} - x_j) 1\{x_j < R_{j-1}\} \\ &= \omega^I \mathbb{E}(x_j - R_{j-1})^+ + \omega_j^W \mathbb{E}(R_{j-1} - x_j)^+. \end{aligned}$$

Lemma 1. For $\boldsymbol{\sigma} \in \mathcal{J}$, $\mathbf{x} \in \mathbb{R}_+^n$,

$$\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x}) = \omega^T \sum_{j=1}^{n+1} \mathbb{E} T_j + \sum_{j=1}^n \mathcal{R}_j(\mathbf{x}).$$

Proof. Recall from (1) that the objective function is defined by

$$\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x}) = \omega^T \sum_{j=1}^{n+1} \mathbb{E} T_j + \omega^I \sum_{j=1}^n \mathbb{E} I_j + \sum_{j=1}^n \omega_j^W \mathbb{E} W_j.$$

As we have that $R_j = W_j + U_{j+1}$, the Lindley recursion becomes

$$I_j = \max\{x_j - R_{j-1}, 0\}, \quad W_j = \max\{R_{j-1} - x_j, 0\}.$$

The stated follows directly. \square

The following lemma helps evaluating $\mathcal{R}_j(\mathbf{x})$ when the random variables R_j are of phase type.

Lemma 2. *Let the non-negative random variable X be of phase type of dimension d , with the initial distribution represented by the row vector $\boldsymbol{\alpha} \in \mathbb{R}_+^d$ and the $d \times d$ transition rate matrix given by V . Then, for any $A, B \in \mathbb{R}$ and $x \geq 0$,*

$$A \mathbb{E}(X - x)^+ + B \mathbb{E}(x - X)^+ = -(A + B) \boldsymbol{\alpha} V^{-1} e^{Vx} \mathbf{1} + B(x + \boldsymbol{\alpha} V^{-1} \mathbf{1}).$$

Proof. Start by observing that

$$A(X - x)1\{X \geq x\} + B(x - X)1\{X < x\} = (A + B)(X - x)1\{X \geq x\} + B(x - X). \quad (6)$$

Denote $\mathbf{v} := -V\mathbf{1}$. Then, using integration by parts, performing the change of variables $z := y - x$, and denoting by $f_X(x)$ the density of X (as given in Appendix A),

$$\begin{aligned} \mathbb{E}(X - x)^+ &= \int_x^\infty f_X(y) (y - x) dy = \int_x^\infty \boldsymbol{\alpha} e^{Vy} \mathbf{v} (y - x) dy \\ &= - \int_0^\infty \boldsymbol{\alpha} V^{-1} e^{V(x+z)} \mathbf{v} dz = \int_0^\infty \boldsymbol{\alpha} e^{V(x+z)} \mathbf{1} dz = -\boldsymbol{\alpha} V^{-1} e^{Vx} \mathbf{1}, \end{aligned}$$

where it is used that a matrix and its matrix exponent commute. Now using the standard result that $\mathbb{E}X = -\boldsymbol{\alpha} V^{-1} \mathbf{1}$ (which of course also follows from inserting $x = 0$ in the obtained expression for $\mathbb{E}(X - x)^+$), the stated follows. \square

From the above we conclude that we are left with finding the phase-type description of the random variables R_j . Let the phase type description \bar{U}_j of U_j be given by the row vector $\boldsymbol{\alpha}_j \in \mathbb{R}_+^{d_j}$ and the $d_j \times d_j$ transition rate matrix given by V_j , where $\mathbf{v}_j := -V_j \mathbf{1}$ (where the dimension of this all-ones vector $\mathbf{1}$ is d_j). Define the matrices $V^{(j)}$ in the following recursive manner. Let $D_j := d_1 + \dots + d_j$ be the sum of the dimensions of phase-type random variables \bar{U}_1 up to \bar{U}_j , and let $\mathbb{O}_{m,n}$ be an all-zeroes matrix of dimension $m \times n$. With $V^{(1)} = V_1$, we recursively define

$$V^{(j)} := \begin{pmatrix} V^{(j-1)} & \tilde{V}_j \\ \mathbb{O}_{d_j, D_{j-1}} & V_j \end{pmatrix},$$

where, for $j = 2, 3, \dots$,

$$\tilde{V}_j = \begin{pmatrix} \mathbb{O}_{D_{j-2}, d_j} \\ -V_{j-1} \mathbf{1} \boldsymbol{\alpha}_j \end{pmatrix};$$

observe that the dimension of $-V_{j-1} \mathbf{1} \boldsymbol{\alpha}_j$ is $d_{j-1} \times d_j$, so that \tilde{V}_j has dimension $D_{j-1} \times d_j$ and $V^{(j)}$ has dimension $D_j \times D_j$.

We compute the objects $F_j(y)$ and $\mathbf{P}_j(y)$ in the following recursive manner. Define, for $y \geq 0$,

$$F_j(y) := 1 - \mathbf{P}_j(y) \mathbf{1},$$

where $\mathbf{P}_1(y) := \boldsymbol{\alpha}_1 e^{V^{(1)}y}$ and, for $j = 2, 3, \dots$,

$$\mathbf{P}_j(y) := (\mathbf{P}_{j-1}(x_j), \boldsymbol{\alpha}_j F_{j-1}(x_j)) e^{V^{(j)}y},$$

being a D_j -dimensional row vector. Mimicking the procedure developed in [38], the following result can be derived.

Lemma 3. *For all $y \geq 0$,*

$$\mathbb{P}(R_j \leq y) = F_j(y) = 1 - \mathbf{P}_j(y) \mathbf{1}.$$

The lemma directly entails that R_j has a phase-type distribution of dimension D_j , with the initial distribution represented by the row vector

$$\boldsymbol{\alpha}_{(j)} := (\mathbf{P}_{j-1}(x_j), \boldsymbol{\alpha}_j F_{j-1}(x_j)) \in \mathbb{R}^{D_j}$$

and the $D_j \times D_j$ transition rate matrix given by $V^{(j)}$. Combining the above three lemmas, we find the closed form expression for the objective function $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$ as presented in Theorem 1. Here, we use that the component corresponding to the idle time is

$$\sum_{j=1}^n \omega^I \left(x_j + \boldsymbol{\alpha}_{(j)} (V^{(j)})^{-1} \mathbf{1} - \boldsymbol{\alpha}_{(j)} (V^{(j)})^{-1} e^{V^{(j)} x_j} \mathbf{1} \right),$$

whereas the component due to waiting times becomes

$$- \sum_{j=1}^n \omega_j^W \boldsymbol{\alpha}_{(j)} (V^{(j)})^{-1} e^{V^{(j)} x_j} \mathbf{1}.$$

With Theorem 1 at our disposal, the numerical evaluation of $\mathcal{L}(\boldsymbol{\sigma}, \mathbf{x})$ has become a standard task. It requires matrix multiplication, matrix inversion, and the evaluation of matrix exponentials, which are all standard procedures in virtually any numerical software package. For the evaluation of the matrix exponential, it is a computational advantage that the matrices $V^{(j)}$ are triangular so that the eigenvalues appear on their diagonals.

APPENDIX C. MODIFIED TRAVELING SALESMAN PROBLEM

The MTSP heuristic is from [43]. The idea is to modify the cost of traveling between any pair of clients by taking both the traveling cost as well as the appointment cost (idle and waiting time) into account. As a proxy, the appointment cost for some client j is based on the visit of only its preceding client i :

$$C_{ij}(x_j) = \omega_j^W \mathbb{E}(U_{ij} - x_j)^+ + \omega^I \mathbb{E}(x_j - U_{ij})^+, \quad (7)$$

where $U_{ij} = B_i + T_{ij}$. Here x_j represents the inter-appointment time between clients i and j . Due to the newsvendor model, we directly obtain that (7) is minimized by

$$x_j^* = F_{ij}^{-1} \left(\frac{\omega_j^W}{\omega_j^W + \omega^I} \right),$$

with $F_{ij}^{-1}(\cdot)$ denoting the inverse of U_{ij} . The traveling costs are then modified as $\hat{c}_{ij} = \mathbb{E}T_{ij} + C_{ij}(x_j^*)/\omega^I$. First, the route $\boldsymbol{\sigma}^{\text{MTSP}}$ is determined as the solution of the asymmetric TSP with travel costs \hat{c}_{ij} ; see Algorithm 1 of [43] for an algorithmic outline. Second, the optimal appointment schedule is then obtained by minimizing the inter-appointment times $\min_{\mathbf{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{\sigma}^{\text{MTSP}}, \mathbf{x})$.

It remains to specify the appointment costs $C_{ij}(x_j)$ in (7). Due to (6), we may rewrite (7) as

$$C_{ij}(x_j) = (\omega_j^W + \omega^I) \mathbb{E}(U_{ij} - x_j)^+ + \omega^I x_j - \omega^I \mathbb{E}U_{ij},$$

such that we only need $\mathbb{E}(U_{ij} - x_j)^+$. As in Subsection 4.1, we use the phase-type description \bar{U}_{ij} of U_{ij} . Now, for $\text{SCV}(U_{ij}) \geq 1$, such that $\bar{U}_{ij} \sim H_2(\mu_1, \mu_2, p)$ for some parameters μ_1, μ_2 , and p ,

$$\mathbb{E}(\bar{U}_{ij} - x_j)^+ = \frac{p}{\mu_1} e^{-\mu_1 x_j} + \frac{1-p}{\mu_2} e^{-\mu_2 x_j}.$$

Otherwise, for $\text{scv}(U_{ij}) < 1$, such that $\bar{U}_{ij} \sim E_K(\mu, p)$ for some parameters μ and p , it follows after some lengthy calculations that

$$\mathbb{E}(\bar{U}_{ij} - x_j)^+ = \frac{k-p-\mu x_j}{\mu(k-2)!} \Gamma(k-1, \mu x_j) + \frac{k-p}{\mu(k-1)!} (\mu x_j)^{k-1} e^{-\mu x_j},$$

where $\Gamma(k, t)$ denotes the incomplete Gamma integral $\int_t^\infty z^{k-1} e^{-z} dz$.

APPENDIX D. ACCURACY OF HYBRID APPROXIMATION

In this section, we consider again the appointment scheduling component of the RAS, assuming a fixed route σ . When searching for good appointment schedules, we need to find a balance between speed and accuracy, where speed is particularly important due to the iterative nature of the search procedure for the routing component. Below, we provide a rationale for opting for the hybrid approximation over the phase-type method and the heavy traffic approximations.

As indicated in Subsection 4.1, determining optimized inter-appointment times using the phase-type method $\mathcal{L}_{\text{ph}}(\sigma)$ becomes prohibitively slow. Also, observe that $\mathcal{L}_{\text{ph}}(\sigma)$ corresponds to the true optimal objective function $\mathcal{L}(\sigma) := \min_{\mathbf{x} \in \mathbb{R}_+^n} \mathcal{L}(\sigma, \mathbf{x})$ in case U_j follows a mixed Erlang E_K or hyperexponential H_2 distribution. Hence, we wish to investigate how the hybrid $\mathcal{L}_{\text{hyb}}(\sigma)$ and heavy-traffic $\mathcal{L}_{\text{ht}}(\sigma)$ approximations perform compared to the true optimum $\mathcal{L}(\sigma)$ for a variety of routes σ in case of phase-type U_j .

The performance of the hybrid and heavy-traffic approximations are quantified, respectively, through the relative errors

$$\Delta_{\text{hyb}} = \frac{\mathcal{L}_{\text{hyb}}(\sigma) - \mathcal{L}(\sigma)}{\mathcal{L}(\sigma)} \cdot 100\%, \quad \Delta_{\text{ht}} := \frac{\mathcal{L}_{\text{ht}}(\sigma) - \mathcal{L}(\sigma)}{\mathcal{L}(\sigma)} \cdot 100\%.$$

To consider a variety of instances, we determine the errors for 200 randomly generated scenarios. The mean of each of the U_j is sampled uniformly from $[0.5, 1.5]$ and the SCV is sampled uniformly from $[0.2, 1.8]$. All instances have 40 clients, whereas $\omega^I = 0.8$ and $\omega_j^W = 0.2$ for every j .

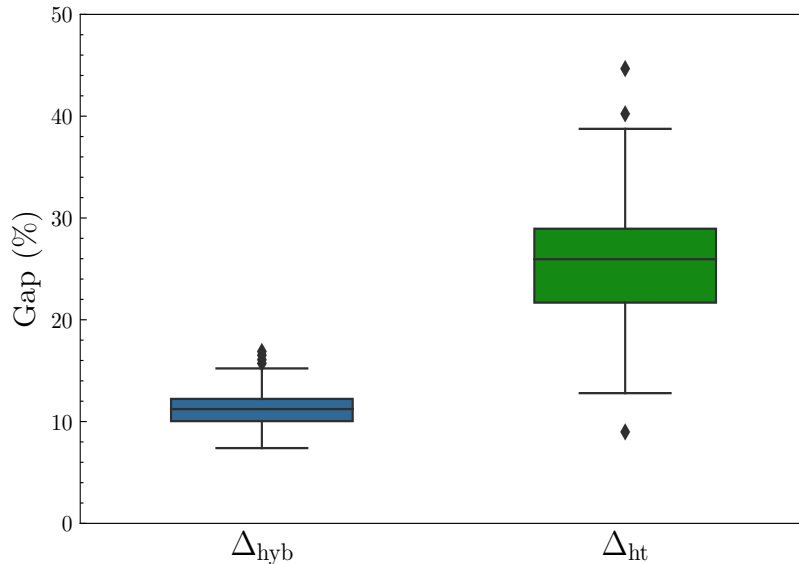


FIGURE 5. Box plot of Δ_{hyb} and Δ_{ht} for 200 random scenarios.

Figure 5 displays the box plots of Δ_{hyb} and Δ_{ht} . We see that we lose some performance, when inter-appointment times are scheduled based on heavy traffic. For instance, the median of Δ_{hyb} is 11.2%, and the median of Δ_{ht} is even 25.9%. To assess the impact of the route σ , it is more important that the

difference with $\mathcal{L}(\sigma)$ is relatively stable. After all, for the routing component, we wish to compare the absolute difference in objective function between any two alternative routes σ^a and σ^b ; this difference will not be affected by a constant error in the objective function. This is well achieved by $\mathcal{L}_{\text{hyb}}(\sigma)$ given the narrow boxplot; the interquartile range (IQR) is only 2.18% for Δ_{hyb} , whereas the IQR is 7.29% for Δ_{ht} .

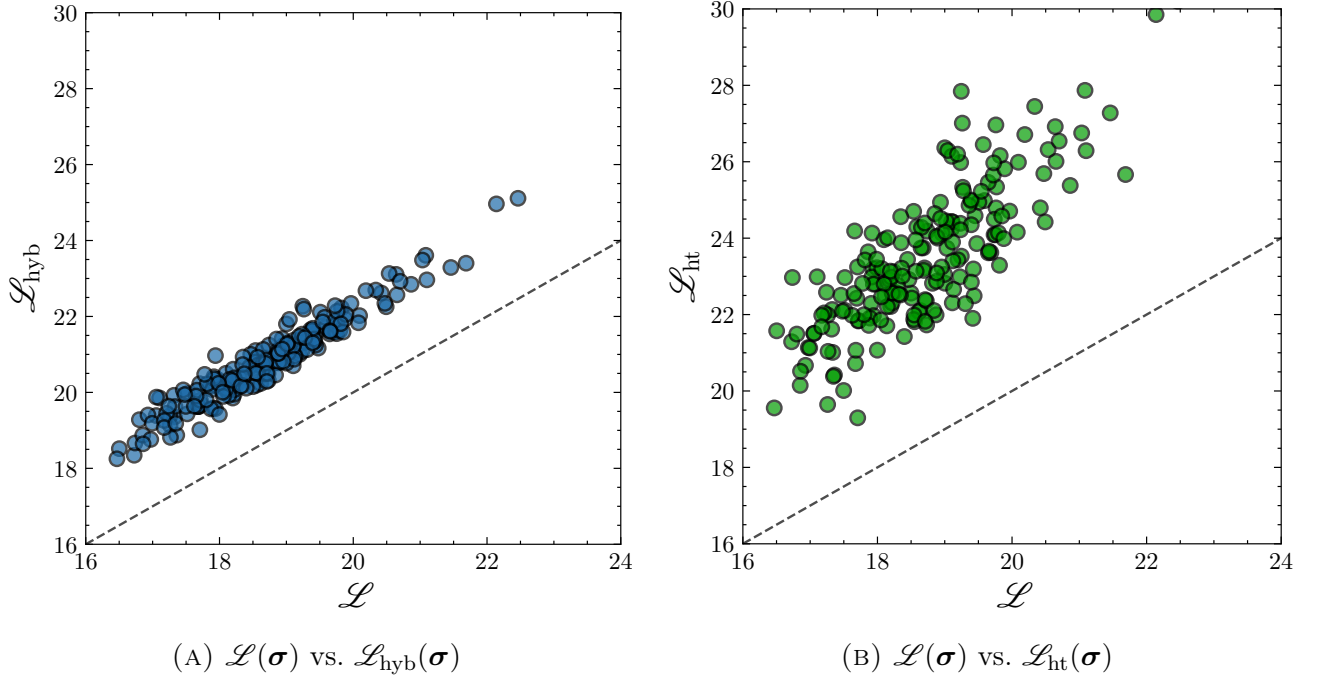


FIGURE 6. Scatterplots of the objective functions for 200 random scenarios.

Figure 6 presents the objective function for the 200 random scenarios using scatterplots. In line with the boxplots, Figure 6a clearly shows a strong linear relation between the hybrid approximation $\mathcal{L}_{\text{hyb}}(\sigma)$ and the true optimal $\mathcal{L}(\sigma)$ with little ‘noise’. The relation between $\mathcal{L}_{\text{ht}}(\sigma)$ and $\mathcal{L}(\sigma)$ in Figure 6b is also clearly present, but not as strong.

No.	Scenario (σ)	$\mathcal{L}(\sigma)$	$\mathcal{L}_{\text{hyb}}(\sigma)$	$\mathcal{L}_{\text{ht}}(\sigma)$	Δ_{hyb}	Δ_{ht}
1	3 2 2 2 2 2 2 1	8.35	10.15	11.93	21.6	42.8
2	1 3 2 2 2 2 2 2	8.07	9.54	10.02	18.2	24.1
3	1 2 2 3 2 2 2 2	8.00	9.41	9.59	17.6	21.1
4	1 2 2 2 2 3 2 2	7.92	9.31	9.38	17.6	18.4
5	1 2 2 2 2 2 3 2	7.83	9.26	9.30	18.2	18.7
6	1 2 2 2 2 2 2 3	7.65	9.17	9.24	19.9	20.7

TABLE 3. Results of different scenarios with $n = 40$ locations including depot, and their respective SCV’s that are arranged in batches of 5 where, 1=0.3, 2=0.9, 3=1.5, and $\omega^1 = 0.8$.

Finally, we carried out some experiments to investigate how the objective function behaves with respect to the SVF rule. As alluded to in the introduction, the intuition behind SVF is that highly variable service times are more likely to cause (excessive) delays, which can have a ‘cascade’ affect for all subsequent clients; scheduling clients with less variable service times earlier should result in smaller overall waiting

and idle times. Again, we consider $n = 40$ clients and $\omega^I = 0.8$ and $\omega_j^W = 0.2$ for every j . The 40 clients are divided into 8 batches with each batch having 5 clients with the same SCV. In Table 3, we considered six scenarios, where the numbers 1, 2 and 3 represent batches of clients with an SCV of 0.3, 0.9, and 1.5, respectively. Scenario 6 corresponds to SVF and has indeed the lowest objective function. In line with intuition, the objective function becomes lower when the batch of clients with the larger SCV is later in the schedule; this holds for all $\mathcal{L}(\sigma)$, $\mathcal{L}_{\text{hyb}}(\sigma)$, and $\mathcal{L}_{\text{ht}}(\sigma)$. Moreover, the gap Δ_{hyb} for the hybrid approximation is rather stable, whereas the heavy traffic approximation has a much larger gap Δ_{ht} for scenario 1.

To summarize, from our experiments we observe that the hybrid approximation is a suitable approach to compare alternative routes with optimized inter-appointment times. The variability in the heavy traffic approximation seems to large to guide the search process for the RAS.

REFERENCES

- [1] A. Ahmadi-Javid, Z. Jalali, and K. J. Klassen. Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1):3–34, 2017.
- [2] A. Ala and F. Chen. Appointment scheduling problem in complexity systems of the healthcare services: A comprehensive review. *Journal of Healthcare Engineering*, 2022.
- [3] S. Asmussen. *Applied Probability and Queues*, volume 2. Springer, New York, 2003.
- [4] F. Camacho, R. Anderson, A. Safrit, A. S. Jones, and P. Hoffmann. The relationship between patient’s perceived waiting time and office-based practice satisfaction. *North Carolina Medical Journal*, 67(6):409–413, 2006.
- [5] T. Cayirli and E. Veral. Outpatient scheduling in health care: A review of literature. *Production and Operations Management*, 12(4):519–549, 2003.
- [6] W. J. Cook. In *Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2015.
- [7] K. Dalmeijer and R. Spliet. A branch-and-cut algorithm for the time window assignment vehicle routing problem. *Computers & Operations Research*, 89:140–152, 2018.
- [8] B. Denton and D. Gupta. A sequential bounding approach for optimal appointment scheduling. *IIE Transactions*, 35(11):1003–1016, 2003.
- [9] G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.
- [10] R. B. Fetter and J. D. Thompson. Patients’ waiting time and doctors’ idle time in the outpatient setting. *Health Services Research*, 1(1):66–90, 1966.
- [11] M. Gendreau and J. Y. Potvin. *Handbook of Metaheuristics*, volume 2. Springer, New York, 2010.
- [12] D. Gupta and B. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions*, 40(9):800–819, 2008.
- [13] K. Helsgaun. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12:966–980, 2017.
- [14] C.-J. Ho and H.-S. Lau. Minimizing total cost in scheduling outpatient appointments. *Management Science*, 38(12):1750–1764, 1992.
- [15] C. Hu, J. Lu, X. Liu, and G. Zhang. Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. *Computers & Operations Research*, 94:139–153, 2018.
- [16] G. C. Kaandorp and G. Koole. Optimal outpatient appointment scheduling. *Health Care Management Science*, 10:217–229, 2007.
- [17] K. J. Klassen and R. Yoogalingam. Improving performance in outpatient appointment services with a simulation optimization approach. *Production and Operations Management*, 18(4):447–458, 2009.
- [18] K. J. Klassen and R. Yoogalingam. Appointment scheduling in multi-stage outpatient clinics. *Health Care Management Science*, 22:229–244, 2019.
- [19] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.
- [20] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational Research*, 22(3):2033–2062, 2022.

- [21] A. Kuiper. *Appointment Scheduling in Healthcare*. PhD thesis, Universiteit van Amsterdam, 2016.
- [22] A. Kuiper, B. Kemper, and M. Mandjes. A computational approach to optimized appointment scheduling. *Queueing Systems*, 79:5–36, 2015.
- [23] A. Kuiper and R. H. Lee. Appointment scheduling for multiple servers. *Management Science*, 68(10):7422–7440, 2022.
- [24] A. Kuiper, M. Mandjes, and J. de Mast. Optimal stationary appointment schedules. *Operations Research Letters*, 45(6):549–555, 2017.
- [25] A. Kuiper, M. Mandjes, J. de Mast, and R. Brokkelkamp. A flexible and optimal approach for appointment scheduling in healthcare. *Decision Sciences*, 54(1):85–100, 2023.
- [26] J. Li, F. Wang, and Y. He. Electric vehicle routing problem with battery swapping considering energy consumption and carbon emissions. *Sustainability*, 12(24), 2020.
- [27] D. Pisinger and S. Ropke. Large Neighborhood Search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 272, pages 99–127. Springer International Publishing, 2019.
- [28] W. B. Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821, 2019.
- [29] A. Ruszczyński and A. Shapiro. Stochastic programming models. *Handbooks in Operations Research and Management Science*, 10:1–64, 2003.
- [30] P. Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In G. Goos, J. Hartmanis, J. van Leeuwen, M. Maher, and J.-F. Puget, editors, *Principles and Practice of Constraint Programming – CP98*, volume 1520, pages 417–431. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
- [31] 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. *Transportation Research Part E: Logistics and Transportation Review*, 128:52–95, 2019.
- [32] R. Spliet and A. F. Gabor. The time window assignment vehicle routing problem. *Transportation Science*, 49(4):721–731, 2015.
- [33] S. Srinivas and A. R. Ravindran. Optimizing outpatient appointment system using machine learning algorithms and scheduling rules: A prescriptive analytics framework. *Expert Systems with Applications*, 102:245–261, 2018.
- [34] H. C. Tijms. *Stochastic Modelling and Analysis: A Computational Approach*. John Wiley & Sons, Chichester & New York, 1986.
- [35] M. Y. Tsang and K. S. Shehadeh. Stochastic optimization models for a home service routing and appointment scheduling problem with random travel and service times. *European Journal of Operational Research*, 307(1):48–63, 2023.
- [36] P. M. Vanden Bosch and D. C. Dietz. Scheduling and sequencing arrivals to an appointment system. *Journal of Service Research*, 4(1):15–25, 2001.
- [37] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020.
- [38] P. P. Wang. Optimally scheduling N customer arrival times for a single-server system. *Computers & Operations Research*, 24(8):703–716, 1997.
- [39] J. D. Welch and N. T. J. Bailey. Appointment systems in hospital outpatient departments. *The Lancet*, 259(6718):1105–1108, 1952.
- [40] N. A. Wouda and L. Lan. ALNS: a Python implementation of the adaptive large neighbourhood search metaheuristic. *Journal of Open Source Software*, 8(81):5028, 2023.
- [41] C. Zacharias and T. Yunes. Multimodularity in the stochastic appointment scheduling problem with discrete arrival epochs. *Management Science*, 66(2):744–763, 2020.
- [42] Y. Zhan and G. Wan. Vehicle routing and appointment scheduling with team assignment for home services. *Computers & Operations Research*, 100:1–11, 2018.
- [43] Y. Zhan, Z. Wang, and G. Wan. Home service routing and appointment scheduling with stochastic service times. *European Journal of Operational Research*, 288(1):98–110, 2021.