Simheuristic and learnheuristic Algorithms for the Temporary-Facility Location and Queuing Problem during during Population Treatment or Testing Events

Christopher Bayliss[1,*] Javier Panadero[2,†]
[1] Management School, University of Liverpool, Liverpool L69 7ZH, UK
[2]Dept. of Management, Universitat Politècnica de Catalunya – BarcelonaTech, Barcelona, 08028, Spain; javier.panadero@upc.edu

Abstract

Epidemic outbreaks, such as the one generated by the coronavirus disease, have raised the need for more efficient healthcare logistics. One of the challenges that many governments have to face in such scenarios is the deployment of temporary medical facilities across a region with the purpose of providing medical services to their citizens. This work tackles this temporary-facility location and queuing problem with the goals of minimizing costs, the expected completion time, population travel and waiting times. The completion time for a facility depends on the numbers assigned to those facilities as well as stochastic arrival times. This work proposes a learnheuristic algorithm to solve the facility location and population assignment problem. Firstly a machine learning algorithm is trained using data from a queuing model (simulation module). The learnheuristic then constructs solutions using the machine learning algorithm to rapidly evaluate decisions in terms of facility completion and population waiting times. The efficiency and quality of the algorithm is demonstrated by comparison with exact and simulation-only (simheuristic) methodologies. A series of experiments are performed which explore the trade offs between solution cost, completion time, population travel and waiting times.

Keywords: Facilities planning and design, Queuing, sim-learnheuristics, Simulation, Machine learning.

1 Introduction

Logistics plays a vital role in a large number of contexts. Apart from being massively a tool of study in supply chain systems in the last decades, it has emerged in substantially different fields, such as healthcare (Ageron et al., 2018; Moons et al., 2019), disaster management (Jahre et al., 2007; Oksuz and Satoglu, 2020), telecommunication, and so on. Many logistics activities face the challenge of defining a set of potential facilities for supplying a set of clients subject to many constraints, such as limited capacity and inventory. Depending on the application context, these clients assume different classes, which vary from consumers themselves to physical locations –e.g., depots and health centers. In general, this class of facility location problems (FLPs) regards to the taking of decisions to decide how many facilities must be opened and where to locate them in order to minimize the total transportation cost of the network. Moreover, depending on the application context, these decisions may be difficult to change in the short-term, which highlights the importance of respective choices.

The location of facilities is crucial in healthcare systems. In such scenarios, the importance of strategically selecting facilities is related to both mortality and morbidity rates. For example, by opening a small number of

*Corresponding author: Christopher.Bayliss@liverpool.ac.uk
†Coauthor: jpanaderom@uoc.edu

facilities or poorly locating them, the resulting network leads to an increment in the number of deaths and diseases in the community (Daskin and Dean, 2005). Therefore, incorrect facility location decisions have a serious impact not only on the operational cost but also in the target community and customer service metrics (Ahmadi-Javid et al., 2017), which are of greater importance when applied to healthcare. Especially in healthcare, these particularities encourage the use of simulation tools, such as discrete-event simulation (DES) techniques, to support the forecasting and assessing of the potential impact of changes on patient flow, allocation needs, and, finally, to investigate the complex relationships among different system variables. According to Jacobson et al. (2006), these simulation tools allow managers to identify and explore alternative ways to reconfigure existing systems, in order to improve their performance or design, while keeping the existing one unchanged.

In this work, we consider an epidemic situation in which a set of facilities must be strategically established in order to offer proper healthcare treatment to a maximum number of individuals from a vulnerable population. These individuals are allocated to nearby deployed facilities (Figure 1), where they queue for treatment (Figure 2). At each facility, a single queue/server is available to serve the assigned individuals. Note that multiple server facilities can be modeled as multiple single facilities in close proximity. The first patient to arrive is the first to be served –i.e., a first come first serve (FCFS) basis. Regarding the need for some people to receive multiple treatments, from a facility planning perspective only the total number of required treatments needs to be known. The objectives are to minimize: *(i)* the total travel time between residences and nearest facilities; *(ii)* the total time spent queuing by individuals waiting for treatment, and *(iii)* the time required to deliver the vaccine to the population. In other words, we aim to minimize the total time required to deliver the treatments to all of the vulnerable population, i.e., to minimize the makespan. Minimizing travel time and queuing times make the process more convenient and safer for the population, while minimizing the total time required to deliver the vaccine to the population reduces costs. Minimizing total treatment time, or makespan, any ultra-low temperature vaccine storage costs can also be minimized. According to Moons et al. (2019), by improving the efficiency of healthcare logistics activities, such as those addressed in this work, the quality of care is increased and the related costs are reduced. Therefore, by optimizing this whole process, not only will the population benefit from better services, which reduces the probability of deaths or complications, but healthcare centers and governments can offer more efficient services with lower costs.

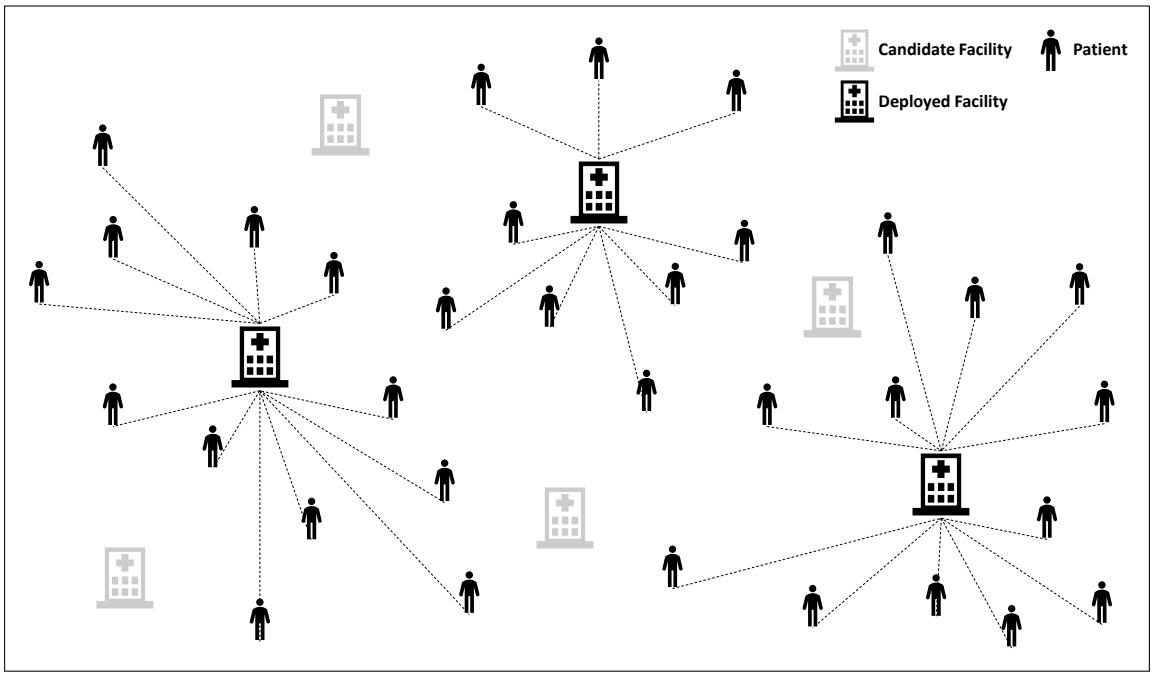

Figure 1: Visual representation of the problem.

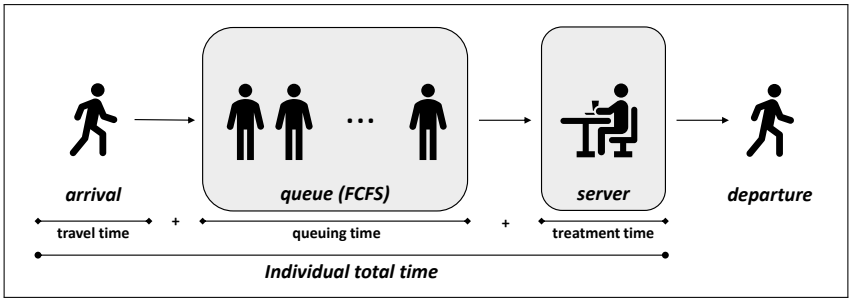

Figure 2: Queuing process at each facility.

This problem of delivering vaccines to a vulnerable population during an epidemic crisis can be formulated as an uncapacitated facility location problem (UFLP) with queuing simulation. Here, "uncapacitated" refers to the fact that each individual facility can treat the whole population, albeit in a relatively long amount of time. It is referred to as a 'temporary' problem due to the particular scenario which is generated during a public healthcare crisis. The UFLP is a popular *NP-hard* problem, formally stated in Balinski (1964a), which has been largely applied to logistics and supply chains. However, by integrating and introducing this problem with queuing strategies in healthcare, a challenging integrated problem application for this type of crisis management is opened up. The contributions of this work include the following. The integrated facility location and queuing problem is formulated and solved for small instances as a mixed integer program. This work provides a black-box optimization framework (learnheuristic) which integrates simulation, heuristic optimization and machine learning for solving large scale instances of the integrated facility location and queuing problem. The learnheuristic is shown to provide results of equal quality to those of the exact approach and a simulation-only (simheuristic) equivalent methodology (see Section 9.3). That is, in this work, the learnheuristic extends the simheuristic framework by making use of a machine learning module in order to improve computational efficiency. This work tackles the issue of solution dependent decision costs that arise in the application of a constructive heuristic solution approach to the integrated facility location and queuing problem. In particular, we outline how machine learning can be used to reduce excessive use of simulation that would be required for the accurate evaluation of dynamically changing decisions costs. We propose a method for controlling population waiting times based on the allocation of treatment time windows.

The rest of the paper is arranged as follows: Section 2 presents a literature review on related topics; Section 3 introduces the problem description; Section 4 presents a novel alternative mathematical formulation; Section 5 introduces our methodology; Sections 6, 7 and 8 present the DES, machine learning and optimization components of the learnheuristic respectively; Section 9 analyzes a series of computational experiments and discusses the achieved results; finally, Section 10 highlights the main conclusions of this work and proposes some lines for future research.

2 Related Work

2.1 The Facility Location Problem

The FLP was initially introduced by Balinski (1964b), referred to as the plant location problem. In the existing literature, many authors have used exact methods for solving it. In this sense, researchers deeply explored the use of branch-and-bound algorithms. Efroymson and Ray (1966) propose a branch-and-bound algorithm for plant location. They used a compact formulation of the FLP to take advantage of the fact that its linear programming relaxation can be solved by inspection. Erlenkotter (1978) proposes a dual-based branch-and-bound algorithm for solving the uncapacitated facility location problem (UFLP) based on a linear programming dual formulation. An enhanced version of this original algorithm was presented by Körkel (1989). Due to the NP-hard nature of

the problem, it can not be guaranteed that exact methods find high-quality solutions within reasonable computing time for large problems. Thus, approximate methods become a natural choice for solving large-scale and realistic instances in a reasonable amount of time. In that sense, several heuristics and metaheuristics algorithms have been developed to solve the FLP and its variants. Therefore, de Armas et al. (2017) propose a fast savings-based heuristic for solving the UFLP, which is then extended into an iterated local search (ILS) metaheuristic framework. The proposed framework employs an acceptance criterion for accepting worse solutions within a margin limit. Another interesting approach to solve the UFLP is presented by Hakli and Ortacay (2019), where an improved scatter search algorithm to solve the UFLP is proposed. The authors apply different crossover techniques to generate new solutions, and mutation operations to improve these solutions. Consequently, it could overcome different population-based approaches, such as those based on swarm optimization and evolutionary algorithms. Estrada-Moreno et al. (2020) present a metaheuristic algorithm based on the integration of a biased-randomized procedure (Grasas et al., 2016) with an iterated local search (ILS) framework (Lourenço et al., 2019), to efficiently cope with the single-source capacitated facility location problem with (SSCFLP) with soft capacity constraints. The algorithm provides a good balance between efficiency and relative simplicity. Lai et al. (2010) propose a hybrid algorithm based on combining a Benders' decomposition algorithm with a genetic algorithm instead of the costly branch-and-bound method. The computational results reported the algorithm is able to obtain good-quality solutions efficiently. However, the author only compared its performance with the Benders' original algorithm. More recently, Martins et al. (2022) propose an Agile Optimization framework (do C. Martins et al., 2021), which combines a biased-randomized algorithm with parallel programming techniques to provide real-time solutions for the UFLP. This approach allows to react and adapt to fast-changing customer demands in the system, reoptimizing the system every time new information is incorporated into the model.

Usually, the inputs of combinatorial optimization problems are not deterministic in real life. Thus, they are subject to random events, e.g., random failures of some components, stockouts due to random demands, etc. Therefore, Simulation-based optimization approaches are required. In that sense, Correia and Saldanha-da Gama (2019) reviewed the FLP with stochastic components, illustrating different methods that appeared in the literature during the last years for optimizing the FLP under uncertainty. Simheuristics (Castaneda et al., 2022) is a simulation-optimization method based on the combination of simulation with metaheuristics, which has been used for solving efficiency different combinatorial optimization problems with stochastic elements. In this regard, Simheuristics have been used by different authors to solve the UFLP with stochastic demands. de Armas et al. (2017) propose a simheuristic for solving the UFLP with stochastic demands. This approach combines an ILS metaheuristic with Monte Carlo simulation to deal with uncertainty. The authors extended a set of deterministic instances to test the simheuristic algorithm, considering that the demands are random variables that follow probabilistic distribution functions. A similar approach is proposed in Quintero-Araujo et al. (2021), where authors present the SimILS framework, a simheuristic algorithm combining Monte Carlo simulation with a metaheuristic framework to solve the capacitated location routing problem (CLRP) with stochastic demands. Specifically, the simulation is embedded into an ILS metaheuristic that uses different perturbation operators and biased randomization techniques to diversify the search. Other authors have focused on solving the FLP with fuzzy components, which introduces a new layer of complexity. Verma et al. (2010) present a fuzzy theory model for dealing with UFLP with fuzzy demands. Due to the problem's complexity, the model can solve just small instances. A similar approach is tackled in Uno et al. (2010), where authors consider that the demands represent fuzzy random variables. To solve the problem, the authors model the problem as a fuzzy random programming problem using $\alpha - level$ sets for fuzzy numbers, transforming the problem into a deterministic programming problem. Then, they use a tabu search metaheuristic to solve the deterministic model. Our work combines heuristics, simulation and machine learning to solve large stochastic UFLPs quickly.

Other authors consider the inputs deterministic but with dynamic behavior. The main idea is that some inputs (e.g., customers' demands, facility locations, etc.) might depend upon the specific configuration of the solution being built (e.g., assigning a customer to one facility or another might change his / her demand value). Silva

et al. (2021) propose a set of three different linear relaxation-based heuristics (LRH) and an evolutionary heuristic that hybridizes a genetic algorithm with a variable neighborhood descent (GA+VND) to solve the Dynamic Facility Location Problem with Modular Capacities (DFLPM). The objective of this problem consists in determining locations and sizes of facilities to minimize location and demand allocation costs with decisions taken periodically over a planning horizon. Wang et al. (2021) present an approximation algorithm based on a primal-dual scheme to solve the dynamic k-level facility location problem (k-DFLP), which is an extension of the UFLP. Wu et al. (2022) propose a supervised learning-driven (SLD) heuristic to solve the capacitated facility location and production planning (CFLPP) problem. The heuristic uses as features the solution values derived from linear programming relaxation, Dantzig–Wolfe decomposition, and column generation, to derive an offline-learned oracle on the optimal solution patterns.

Long waiting times and congestion in facilities are an important concern of decision makers in FLPs. Hence, the behavior of the facilities – in terms of waiting times – has also been studied in combination with the FLP to make the problem more closely resemble reality, resulting in the queuing facility location problem (QFLP), which combine queuing theory with FLPs. Tavakkoli-Moghaddam et al. (2017) propose a new Pareto-based multi-objective optimization model, called MOVDO, to solve the FLP with congestion and pricing policies, immobile servers and random demands under service capacity, queuing capacity and multiple servers. Chaleshtori et al. (2020) propose a mathematical model that considered jockeying to solve the Multi-layer congested facility location problem (MLCFLPs). The model deliberated a combination of FLP and queuing systems, modeled with the M/M/1 queuing network. Due to the complexity of the nonlinear problem, the solution method was enhanced by an NSGA-II metaheuristic genetic algorithm to solve efficiently large-scale instances. A new structure for displaying chromosomes was introduced to improve the genetic algorithm's performance. Zamani et al. (2022) address the problem of facility location with unreliable servers and impatient customers, where each facility functions as a M/M/1 queuing system. They propose a nonlinear mathematical model and two piece-wise MILP relaxations were presented for this problem. An exact solution method (the branch and bound algorithm) and an approximate solution method (the antlion algorithm) were proposed. In this work simulation modeling is used to model queuing processes, and machine learning is used to predict simulation results in order to avoid large computation times in our heuristic solution methodology.

2.2 Facility Location Decisions in Healtcare

One of such problems in healthcare which copes with facility location decisions is the preventive healthcare facility location problem (PHFLP). According to Gu et al. (2010), preventive healthcare can save lives and contribute to a better quality of life by diagnosing serious medical conditions early. This problem consists of identifying the optimal number and location of preventive healthcare facilities in order to maximize people participation, which is ensured by considering that *(i)* each participating individual would seek services of the closest preventive healthcare facility; *(ii)* the probability of participation to a preventive program decreases with distance; and *(iii)* each open facility needs to have a minimum number of clients (Verter and Lapierre, 2002). In this context, Verter and Lapierre (2002) proposed two exact approaches based on a general-purpose branch-and-bound procedure, which exploit the problem structure during the solution process, to solve the PHFLP, modeled as an extension of the maximal covering location problem (MCLP). The MCLP regards locating a predetermined number of facilities in order to maximize the total coverage. The distance was considered as the major determinant of the population's participation in the system. The relationship between volume and quality of service was measured by allowing more than a predetermined number of clients at each facility, to justify the quality of service and the allocation of public funding. The model was reformulated in order to solve two larger real-life problems, in which the authors suggested better solutions composed of fewer facilities than the current configuration. In contrast with the last work, Zhang et al. (2009) proposed an approximate method to solve the PHFLP, based on a non-appointment system allocation of clients to facilities, according to their preferences and the determination

of the best set of locations. The total time required for receiving the preventive service –travel, waiting, and service– was used as a measure of the accessibility of a healthcare facility, where a single queue is available. Similarly to the previous work, the authors have assumed that clients would seek the services of the facility with minimum expected total time. Consequently, they concluded that the expected number of participants from each population zone decreases with the expected total service time for those zones, and centralizing the total system capacity at locations preferred by clients is more effective than using a larger number of smaller facilities. On the other hand, Gu et al. (2010) proposed a bi-objective model for performing the location optimization, solved by an interchange algorithm, whose basic idea consists in relocating an opened facility to an unused one. This procedure was accelerated by building two new data structures, referred to as 'population group' and 'candidate string', and the authors estimated the traveling distance and time accurately through Google Maps API. Experiments showed that this strategy improved a real-life application in Canada. Finally, Zambrano et al. (2016) proposed an agent-based simulation to study the effect of redesigning the points of access to a hospital complex in Chile. The authors considered different measure metrics to assess these effects, such as pedestrian density maps, saturation areas, areas with high patient flow, pedestrian flow at entrances, and service levels in the access points. As a result, the proposed approach, developed by using an AnyLogic library, has suggested the adding of new access points to improve the original design in terms of service level and pedestrian flow density during peak hours. For a survey in healthcare facility location, readers are referred to Ahmadi-Javid et al. (2017).

Among the many applications of simulation models in healthcare, several studies address their use for reducing waiting times in different systems. For instance, Reese et al. (2017) proposed a DES model to evaluate and enhance the performance of walk-in clinics where patients are served without appointments. The resulting methodology was able to provide realistic predictions for the system behaviour under different scenarios. Another example of using simulation modeling for reducing waiting times in healthcare was addressed by Monks and Meskarian (2017). The authors concluded that reducing waiting times when emergency departments (EDs) are of low performance is hard to get. Nuñez-Perez et al. (2017) have also addressed the analysis of accident and emergency (A&E) departments in their study. The authors proposed a DES, which is posteriorly validated to establish whether it is statistically comparable with the real-world. Several performance indicators, such as improvement in patient waiting times, resource utilization, the design of facilities and healthcare networks, reduction of access time to healthcare services, were computed and analyzed. A survey regarding the use of DES in healthcare can be found in Jun et al. (1999), while other examples of simulation for reducing waiting times are addressed in Rohleder et al. (2011) and Martinez et al. (2016).

According to Lakshmi and Iyer (2013), queuing models consist of another trend that is available to support healthcare decision-makers in the following areas: waiting time and utilization analysis, system design, appointments, and system analysis. These authors also have addressed the use of simulation-based queuing models in healthcare, where such models offer support to researchers on investigating the accuracy of analytic formulations and solutions that have simplified a queuing problem. Moreover, they state that minimizing the time that patients have to wait and maximizing the utilization of the servers or resources are conflicting goals. A survey in queuing theory in healthcare –in the areas of waiting time and utilization analysis, system design, and appointment systems– is found in Fomundam and Herrmann (2007). Silva and Serra (2008) addressed a similar problem as the one studied in this work, where emergency service centers must be located and the demand to those centers must be allocated. However, in their case, they introduced priority levels which represent the urgency of the requested service, resulting in the priority queuing covering location problem (PQCLP).

3 Problem Description

This work considers the medical facility location problem for vaccinating or testing a population during an epidemic. From a set of possible facilities J, a subset of these is to be selected to open ($O \subseteq J$). Each member of the

population (P) has to be assigned to an open facility $j \in O$. Each facility opened incurs an opening cost c_j. The treatment of a population member i at their assigned facility j incurs a cost of e_{ij}. While determining which facilities to open and which facilities to assign each population member to we seek to minimize costs. If our objective is only to minimize costs then the optimal solution may involve opening very few facilities. The drawbacks of such a solution include the following: *(i)* the total treatment time (overall makespan) required to treat the entire population may be very long; *(ii)* the total travel distance for the population may be very large; and *(iii)* queues at the facility may be large as will waiting times. As such we consider the minimization of total time, total travel distance, and waiting times in addition to minimizing costs. Minimizing waiting time reduces the risk of further infections as well as making the process as seamless and convenient as possible for the population. The total treatment time is defined as the time between the treatment of the first member of population and the treatment of the last member of the population. Assuming that treatment begins at the same time at all opened facilities the total treatment time is the maximum of the treatment times of all of the facilities.

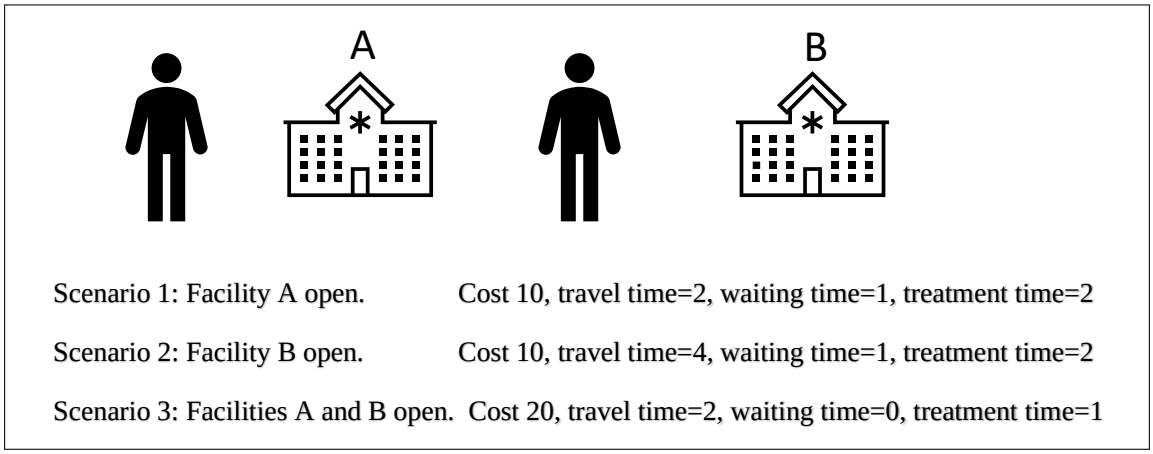

Figure 3: Small numerical example.

For clarification of the problem being addressed, consider the small example in Figure 3. There are two possible single server facilities A and B and a population of 2. There are three possible open facility solutions. In scenario 1 facility A is opened, facility A is located conveniently for both population members in terms of travel time, whereas it is not in scenario 2 were facility B is opened. In both scenarios 1 and 2, only a single facility is opened, which leads to waiting time for one population member, if they arrive at the same time. Also, since a single server facility can only serve one population member at a time, the minimum treatment time is two units of server time. In scenario 3 both facilities are opened, leading to doubled facility opening costs, zero waiting time since each population member can go to a different facility, and a total treatment time of one, since both population members can be served at the same time.

Each population member is also to be assigned to a time window w from the set of possible time windows W. Assigning time windows provides a mechanism for controlling the arrival process and therefore queue sizes and waiting times. It also allows us to address priority constraints, such as vaccinating the most vulnerable first. The exact arrival time of a population member during their assigned time window is still, however, uncertain. The time gap between consecutive time windows may represent a staff break or an overnight period. When a population member arrives at their assigned facility they join the queue for treatment. The facilities operate on a first come first serve (FCFS) basis. The queue size is assumed to unlimited at each facility. Limiting the number of people assigned to each time window and minimizing waiting times provides the mechanism for avoiding very large queues. Each treatment at each facility $j \in O$ consumes an amount of time h_j. Facilities may have the capability to treat people in parallel, they may also have layouts that influence their capacity.

4 Mathematical Formulation

The following models the facility location problem with travel time, waiting time and total treatment time minimization as a mixed integer programming model. The model is solved for a set of scenarios (S), in each of which, the arrival time delay of each member of the population, after the beginning of their assigned time, window is known. In this model population members are assigned to one of the open facilities, as opposed to automatically being assigned to their nearest facility. Objective (1) minimizes costs of opening facilities and treating patients. δ_j is a binary decision variable which indicates whether facility j is selected, c_j is the cost of opening facility j. x_{ijw} is a binary variable indicating which facility j and arrival time window w population member i is assigned to for treatment, e_{ijw} is the cost of treating population member i at facility j in time window w. A weight of $(1-\psi)$ is given to cost considerations. Objective (1) includes penalties that are proportional to the total population transport time T^t, plus total population queuing time T_s^q and overall treatment time T_s^o in each population arrival time scenario. Time penalties are weighted by ψ, furthermore each penalized time has its own weight. α, β and γ are the weights given for penalizing T^t, T_s^q and T^o respectively. Note that since we assume deterministic travel times, the total population travel time T^t is deterministic whereas the total queuing time T_s^q and overall treatment times T_s^o depend on uncertain population arrival times. By varying the values of ψ, α, β and γ we can explore the trade offs between solution cost, travel time, waiting time and treatment time (Section 9.4).

$$\min\ (1-\psi)\left(\sum_{j\in J} c_j\delta_j + \sum_{i\in P}\sum_{j\in J}\sum_{w\in W} e_{ijw}x_{ijw}\right) + \psi\left(\alpha T^t + \frac{1}{|S|}\sum_{s\in S}(\beta T_s^q + \gamma T_s^o)\right). \tag{1}$$

Constraint (2) requires that population members are assigned to one opened medical facility. The set P may represent an entire local population or a prioritized subset of it.

$$\sum_{j\in J}\sum_{w\in W} x_{ijw} = 1,\ \forall i \in P. \tag{2}$$

Constraint (3) ensures that population members are assigned to an opened medical facility.

$$\delta_j \geq \sum_{w\in W} x_{ijw},\ \forall i \in P,\ \forall j \in J. \tag{3}$$

Constraint (4) states that treatment times can only be assigned to population members at their assigned facility. K is the set of treatment times. y_{ijks} is a binary decision variable that takes a value of 1 if population member i is assigned to treatment time k at facility j in population arrival time scenario s. S is the set of arrival time scenarios.

$$\sum_{k\in K} y_{ijks} \leq \sum_{w\in W} x_{ijw},\ \forall i \in P,\ \forall j \in J,\ \forall s \in S. \tag{4}$$

Constraint (5) states that each treatment time at each facility in each scenario can only be assigned to at most 1 member of the population.

$$\sum_{i\in P} y_{ijks} \leq 1,\ \forall k \in K,\ \forall j \in J,\ \forall s \in S. \tag{5}$$

Constraint (6) states that each member of the population is assigned to a treatment time in each scenario.

$$\sum_{j\in J}\sum_{k\in K} y_{ijks} = 1,\ \forall i \in P,\ \forall s \in S. \tag{6}$$

Constraint (7) states that each population member can only be assigned to a treatment time that starts when they arrive or after they arrive at their assigned facility. t_{is}^a is the time that population member i arrives at their assigned facility in population arrival time scenario s. τ_{jk} is the start time of treatment slot k at facility j. Treatment start times can be replicated for the case where facilities have multiple servers, i.e., t_{is}^a can be the same for any two

$i_1, i_2 \in P$. The duration between consecutive treatment slots includes the time to perform the treatment as well as any breaks or safety margin. For the case where treatment slots are of equal duration h_j denotes the duration of treatment slots at facility j, or equivalently, single service time.

$$\sum_{j \in J} \sum_{k \in K} y_{ijks} \tau_{jk} \geq t^a_{is}, \forall i \in P, \ \forall s \in S. \tag{7}$$

Constraint (8) states that each population member arrives at their assigned facility at time t^a_{is}, which depends on the time at which their arrival time window starts (σ_{jw}) plus a stochastic arrival time delay u_{ijws}. u_{ijws} is a random variable representing the time uncertainty of the arrival of population member i in scenario s for their arrival at their assigned facility j. We assume that the uncertain arrival time of a population member occurs within their assigned time window, $u_{ijws} \in [0, m_{jw}]$, modeling early arrivals is a trivial extension. m_{jw} denotes the duration of time window w ($m_{jw} = \sigma_{j(w+1)} - \sigma_{jw}$). The periods between time windows may correspond to staff breaks or overnight periods. If the time per treatment at facility j is h_j on average, then the maximum number of population members that can be assigned to time window w is $\chi_{jw} = \left\lceil \frac{m_j}{h_j} \right\rceil$.

$$t^a_{is} = \sum_{j \in J} \sum_{w \in W} x_{ijw} \left(\sigma_{jw} + u_{ijws} \right), \forall i \in P, \ \forall s \in S. \tag{8}$$

Constraint (9) expresses the total travel time of the population in terms of the customer-facility decisions x_{ijw}, where d_{ij} is the distance between the domicile of population member i and facility j. Note that travel times are deterministic, only facility arrival times are treated as stochastic inputs.

$$T^t = \sum_{i \in P} \sum_{j \in J} \sum_{w \in W} x_{ijw} d_{ij}. \tag{9}$$

Constraint (10) states that the overall time required to treat the population, in any population arrival time scenario s, is greater than any of the assigned treatment slot times. Since we are penalizing the total treatment time in the objective this constraint will be satisfied to equality.

$$T^o_s \geq \sum_{j \in J} \sum_{k \in K} y_{ijks} \tau_{jk}, \forall i \in P, \ \forall s \in S. \tag{10}$$

Constraint (11) states that the overall waiting time, in any population arrival time scenario s, is equal to the sum of all allocated treatment slot times minus the sum of all arrival times (through the associativity of the addition of real numbers).

$$T^q_s = \sum_{i \in P} \left(\left(\sum_{j \in J} \sum_{k \in K} y_{ijks} \tau_{jk} \right) - t^a_{is} \right), \ \forall s \in S. \tag{11}$$

Constraints (12)-(14) specify the binary variables.

$$\delta_j \in \{0,1\}, \ \forall j \in J. \tag{12}$$

$$x_{ijw} \in \{0,1\}, \ \forall i \in P, \ \forall j \in J, \ \forall w \in W. \tag{13}$$

$$y_{ijks} \in \{0,1\}, \ \forall i \in P, \ \forall j \in J, \ k \in K, \ \forall s \in S. \tag{14}$$

5 Learnheuristic Solution Methodology

This work proposes a learnheuristic algorithm (Calvet et al., 2017) for selecting temporary-facility locations and assigning population members to facilities and arrival time windows. Such an approach is useful in cases where decision costs are dynamic. In this case decision costs are dynamic because facility completion times and population waiting times depend on the number of people assigned to a facility, which itself depends on which other facilities are open, which are changed one at a time within a constructive heuristic solution procedure. This type of dynamic decision cost is referred to as "solution dependent decision costs". A consequence of solution dependent decision costs, within each stage of a constructive heuristic methodology, is that the accurate evaluation of decision costs requires a re-evaluation of the each aspect of the solution that will be effected by each possible subsequent decision. In this case, this process will involve multiple sets of repeat simulations, due to the stochastic components of the problem, in particular waiting and treatment times different facilities. In order to reduce the required computation effort, a learnheuristic uses a machine learning algorithm to predict decision costs within each stage of a constructive heuristic instead of simulation.

The learnheuristic algorithm is outlined in Algorithm 1. The initial step of the learnheuristic is to generate input data for the machine learning module (line 3). Following this, the learnheuristic is an iterative algorithm (lines 6 to 26) which generates new "promising" solution in each iteration. Each promising solution is tested using the simulation model (line 19). The simulation returns the average waiting ($t_j^q\,(bestSol)$) and treatment ($t_j^o\,(bestSol)$) times for each open facility in the promising solution ($j \in bestSol$), which are used to calculate the stochastic objective value of the solution. Within each learnheristic iteration multiple runs of a biased randomized constructive algorithm (Section 8) are performed (lines 9 to 17). Decision costs are evaluated within the constructive algorithm using machine learning algorithm predictions (Section 7). The solution with the best estimated objective value from a set of runs of the biased randomized algorithm is deemed the promising solution.

6 Simulation Module

The simulation component is used to obtain information regarding the performance of a solution under stochastic population arrival times. Moreover, the simulation is used to calculate average population waiting times and overall treatment times, i.e, T_s^q and T_s^o of Objective (1), respectively. The simulation model generates random arrival times for each population member within their assigned time window and then simulates the arrival-queuing-treatment process at each open facility. Algorithm 2 outlines the simulation model. Lines 9 to 20 of Algorithm 2 outline the process of generating an arrival scenario for one facility, while lines 21 to 25 describe the simulation of the queuing process and calculation of the average total waiting and treatment times.

For the case of a single server facility with fixed treatment time and uncertain arrival times, *QueueSim* of line 22 of Algorithm 2 corresponds to Algorithm 3, which cycles through the arrivals list in chronological order while updating the total waiting and treatment time.

7 Machine Learning Module

We assume that total waiting time and treatment time are dependent upon the number of patients assigned to a facility (see Figures 4 and 5). As such, our machine learning algorithms will predict expected waiting times and overall treatment times as functions of the numbers of population members assigned to each facility j. The machine learning prediction problems for each facility j can be represented as follows:

$$\tilde{t}_j^q = ML_j^q\left(\sum_{w=1}^{|W|}|P^{jw}|\right). \tag{15}$$

Algorithm 1 Learnheuristic algorithm.

1: Inputs: *Iterations* (number of iterations), *BRRuns* (number of biased randomization iterations used in each learnheuristic iteration), λ (biased randomization parameter), *simulation* (simulation model), *ML* (machine learning algorithms).
2: //Generate input data for the machine learning algorithms.
3: $generateAndCollectData()$ (Algorithm 4)
4: $bestStochasticSol = \emptyset$
5: $bestStochasticObjVal = 0$
6: **for** $i = 1$ to $Iterations$ **do**
7: $bestSol = \emptyset$
8: $bestObjVal = 0$
9: **for** $j = 1$ to $BRRuns$ **do**
10: //Generate a new solution using the biased randomization heuristic.
11: $(sol, objVal) \leftarrow heuristic(\lambda, ML)$ (Algorithm 5)
12: //Update best solution and its value.
13: **if** $objVal > bestObjVal$ **then**
14: $bestSol = sol$
15: $bestObjVal = objVal$
16: **end if**
17: **end for**
18: //Test the best solution in simulation.
19: $\left(\left\{t_j^q(bestSol),\ \forall j \in bestSol\right\}, \left\{t_j^o(bestSol),\ \forall j \in bestSol\right\}\right) = simulation(bestSol)$ (Algorithm 2)
20: //Calculate the stochastic objective value $stochasticObjVal$ using Equation (1).
21: //Update best stochastic solution and its value.
22: **if** $stochasticObjVal > bestStochasticObjVal$ **then**
23: $bestStochasticSol = bestSol$
24: $bestStochasticObjVal = stochasticObjVal$
25: **end if**
26: **end for**
27: Output: $bestStochasticSol$, $bestStochasticObjVal$

$$\tilde{t}_j^o = ML_j^o\left(\sum_{w=1}^{|W|} |P^{jw}|\right). \tag{16}$$

Where $|P_{jw}|$ is the number of population members assigned to time window w at facility j. For any given set of open facilities O, the magnitudes of the sets P^{jw} are uniquely determined according to the nearest-facility-first-available-time-window assignment rule. For each facility one machine learning algorithm is trained for predicting waiting time and another one trained predicting for total treatment time. In this work the machine learning algorithm is a nearest neighbor approach (Witten et al., 2011) with linear interpolation, a similar approach was used by Bayliss et al. (2020) for solving a team orienteering problem with solution dependent decision costs. We recommend that the choice of machine learning algorithm is made on a case by case basis, as more complicated machine learning algorithms become necessary as simulation complexity increases.

7.1 Nearest Neighbor with Linear Interpolation

The data for the nearest neighbor with linear interpolation predictions is obtained from the simulation model outlined in Section 6, data collection is performed before the learnheuristic optimization phase (line 3 Algorithm 1). Let F^j denote the data collected from simulations of facility j. Let f_k^j denote the k^{th} data point in that set. f_k^j is a tuple containing the average total waiting time $\left(q\left(f_k^j\right)\right)$, average total treatment time $\left(o\left(f_k^j\right)\right)$ and number of arriving population members $\left(p\left(f_k^j\right)\right)$ that occurred at facility j. That is, $f_k^j = \left(q\left(f_k^j\right), o\left(f_k^j\right), p\left(f_k^j\right)\right)$, where $q()$, $o()$ and $p()$ are functions that return the total waiting time, overall treatment time and total number of assigned population members of a data point. When waiting time and treatment time predictions are required

Algorithm 2 Simulation of queues at open facilities. $simulation\,(O)$

Inputs: The set of open facilities $O = \left\{ j \in J | \delta_j = 1 \right\}$. The set of population members assigned to each facility in each time window $P^{jw} = \left\{ i \in P | x_{ijw} = 1 \right\}$, $\forall j \in O$, $\forall w \in W$. Number of repeat simulations ($repeatSimulations = |S|$).
//Initialize average waiting time ($t_j^q\,(O)$) and treatment time ($t_j^o\,(O)$) variables for each facility $j \in O$.
$t_j^q\,(O) \leftarrow 0$, $\forall j \in O$
$t_j^o\,(O) \leftarrow 0$, $\forall j \in O$
//Perform repeat simulations.
for $s = 1$ *to* $repeatSimulations$ **do**
  //Consider each open facility.
  for $j \in O$ **do**
    //Reset the list of facility arrival times.
    $arrivalTimes \leftarrow \emptyset$
    for $w \in W$ **do**
      for $i \in P^{jw}$ **do**
        //Sample a random arrival time for population member i in $\left[\sigma_{jw}, \sigma_{j(w+1)}\right]$. $M_{jw}\,()$ denotes the quantile function (F^{-1}) for the distribution of arrival times at facility j, whatever that may be, in arrival time window w.
        $t_{is}^a = \sigma_{jw} + M_{jw}\,(random(0,1))$
        //Add their arrival time to the arrival time list.
        $arrivalTimes \leftarrow (arrivalTimes, t_{is}^a)$
      end for
    end for
    //Sort the list of arrival times in ascending order.
    $Sort\,(arrivalTimes, ascending)$
    //Simulate the queue at facility j for the current arrival time list $arrivalTimes$.
    $(waitingTime, treatmentTime) \leftarrow QueueSim_j\,(arrivalTimes)$ (Algorithm 3)
    //Update average waiting time and average treatment time.
    $t_j^q\,(O) \leftarrow t_j^q\,(O) + \frac{waitingTime}{repeatSimulations}$
    $t_j^o\,(O) \leftarrow t_j^o\,(O) + \frac{treatmentTime}{repeatSimulations}$
  end for
end for
Output: $\left(\left\{ t_j^q\,(O)\,,\ \forall j \in O \right\},\ \left\{ t_j^o\,(O)\,,\ \forall j \in O \right\} \right)$

for facility j when $\mathfrak{p}$ population members are assigned to that facility, we firstly check if the data set F^j contains a data point with a matching number of assigned population members. If data point $f_{k'}^j$ is found to be that match, then the predicted waiting time and treatment time are $q\left(f_{k'}^j\right)$ and $o\left(f_{k'}^j\right)$ respectively. When there is no exact match, we identify the two data points with the closest matching assigned population members above ($f_{k'}^j$) and below ($f_{k''}^j$) $\mathfrak{p}$. $f_{k'}^j = \underset{f_k^j \in F^j}{\arg\min} \left\{ p\left(f_k^j\right) > \mathfrak{p} \right\}$ and $f_{k''}^j = \underset{f_k^j \in F^j}{\arg\max} \left\{ p\left(f_k^j\right) < \mathfrak{p} \right\}$. The linearly interpolated prediction for waiting time is calculated using Equation (17).

$$\tilde{t}_j^q = q\left(f_{k'}^j\right) + \left(\frac{\mathfrak{p} - p\left(f_{k'}^j\right)}{p\left(f_{k''}^j\right) - p\left(f_{k'}^j\right)} \right) \left(q\left(f_{k''}^j\right) - q\left(f_{k'}^j\right) \right). \tag{17}$$

The treatment time prediction is calculated in the same way by replacing q with o.

7.2 Data Sampling with a Limited Simulation Budget

As described above, the data for the nearest neighbour with linear interpolation predictions are collected in the initial phase of the proposed approach. In this phase we have a budget of simulation runs B_j for each facility. The size of the simulation budget directly influences run time, since the simulation is time-expensive due to its

Algorithm 3 Queue simulation of facility j. $QueueSim_j(arrivalTimes)$

Inputs: $arrivalTimes_i$, $\forall i \in P^{jw}$, $\forall w \in W$ the arrival times of each assigned population member sorted in ascending order.
//Initialise clock time and total waiting time.
$clockTime \leftarrow 0$, $waitingTime \leftarrow 0$
//Cycle through the arrivals list.
for $t \in arrivalTimes$ **do**
    $waitingTime \leftarrow waitingTime + \max(0,\ clockTime - t)$
    $clockTime \leftarrow \max(clockTime,\ t) + h_j$
end for
Output: $(waitingTime,\ clockTime)$

relatively high computational requirements. We have a choice on how this budget is spent, that of the number of simulations used to evaluate each sample point, where a sample point is a number of population members that may be assigned to the facility. Let H_j, denote the number of sample points that will be generated, then let the number of repeat simulations used to evaluate each sample point be $R_j = \lceil \frac{B_j}{H_j} \rceil$. We then sample H_j possible numbers of assigned population members at equal intervals between 1 and P_{max} (the total population size). This data generation and collection process is represented in Algorithm 4.

Algorithm 4 Data generation and collection. $generateAndCollectData()$

Inputs: H_j, R_j, J, P_{max}.
//Generate and collect data samples for all facilities $j \in J$.
for $j \in J$ **do**
    //Initialise data sets, which facility is open and the number of repeat simulations per sample point.
    $F^j \leftarrow \emptyset$
    $O \leftarrow \{j\}$
    $repeatSimulations \leftarrow R_j$
    for $i = 1$ to H_j **do**
        //Set the number of assigned population members.
        $|P| \leftarrow \left\lfloor 1 + (i-1)\left(\frac{P_{max}-1}{h_j-1}\right) \right\rfloor$
        //Run the simulation.
        $\left(\left\{t_j^q(O)\right\},\ \left\{t_j^o(O)\right\}\right) \leftarrow simulation(O)$ (Algorithm 2)
        //Collect the data.
        $F^j \leftarrow F^j \cup \left\{\left(t_j^q(O), t_j^o(O),\ |P|\right)\right\}$
    end for
end for
Output: F^j, $\forall j \in J$

8 Optimization Module

Our approach is based on the constructive heuristic proposed by **?**. It starts from an initial solution in which all facilities are open. It then proceeds to close facilities until the savings of closing any other facility is negative. This procedure is outlined in Algorithm 5.

Candidate facilities are sorted in decreasing order of their associated savings value, then a list position index is selected according to a geometric distribution (line 10 Algorithm 5). In that equation, $|+ve\ savings|$ denotes the number of remaining open facilities which have a positive savings value. For $0 < \lambda \leq 1$ the geometric distribution assigns higher selection probabilities to facilities with higher savings values. λ is the probability that facility associated with the highest saving is selected and then closed. The geometric distribution is a natural choice for biased randomization, since the parameter λ provides a single lever controlling the balance between search diversification and intensification.

Algorithm 5 Biased randomized constructive heuristic. $heuristic(\lambda, ML)$

Inputs: λ, ML (machine learning algorithm).
//Set all facilities as open and set the initial objective value (the same every time).
$O \leftarrow J$
$objVal \leftarrow initialObjVal$
//Initialize the savings list (same every time).
$saving_j \leftarrow initialSaving_j$, $\forall j \in O$
//Select facilities to close until the objective cannot be improved.
while $\exists\ saving_{j \in O} \geq 0$, **do**
    //select a facility to close randomly from the sorted savings list according to a geometric distribution.
    $index = Mod\left(\left\lfloor \frac{log(uniRand(0,1))}{log(1-\lambda)} \right\rfloor, |+ve\ saving|\right)$
    $O \leftarrow O \setminus \{O_{index}\}$
    $objVal \leftarrow objVal - saving_{index}$
    //Update the savings values for closing the remaining open facilities.
    $updateSavings(O_{index})$ (Algorithm 6)
end while
Output: $(O,\ objVal)$

8.1 Savings Calculation

For the case of Objective (1) the saving for removing facility j from the set of open facilities O is given by Equation 18.

$$\begin{aligned} saving_j \quad &= (1-\psi)\left(c_j - \sum_{i \in P^j}\left(e_{ij} - e_{iN(i)}\right)\right) \\ &-\psi\left(\alpha \sum_{i \in P^j}\left(d_{ij} - d_{iN(i)}\right) + \beta\left(\sum_{k \in O} t_k^q(O) - \sum_{k \in O \setminus \{j\}} t_k^q(O \setminus \{j\})\right)\right. \\ &\left.+\ \gamma\left(\max_{k \in O}\left(t_k^o(O)\right) - \max_{k \in O \setminus \{j\}}\left(t_k^o(O \setminus \{j\})\right)\right)\right) \end{aligned} \tag{18}$$

Here, P^j denotes the set of population members currently assigned to facility j. $N(i)$ denotes the open facility that population member i is allocated to when facility j is closed, given which are the other remaining open facilities. In this case population members are always assigned to the nearest open facility, as a heuristic approach for reducing travel times. The first term of Equation (18) evaluates the cost savings associated with opening facility j. The second term accounts for savings associated with travel time, waiting times and total treatment time. $t_j^q(O)$ denotes the expected waiting time at facility j when the set of open facilities is O and $t_j^o(O)$ denotes the total expected treatment time at facility j when the set of open facilities is O. A similar but complementary equation exists for the saving associated with opening a facility. In this work $t_j^q(O)$ and $t_j^o(O)$ ($j \in O$) are estimated from the simulation model given in Algorithms 2 and 3, making this savings calculation very computationally expensive if used within the heuristic framework introduced in de Armas et al. (2017), since repeat simulations would have to be implemented for each facility every time a facility is added or removed from the current solution in order to estimate the savings associated with each possible subsequent decision. As a tractable alternative we propose to adopt a learnheuristic approach in which the values of $t_j^q(O)$ and $t_j^o(O)$ are predicted from machine learning algorithms trained in an initial training phase.

8.2 Updated Savings when Closing a Facility

Every time a facility is closed the population members assigned to that facility are assigned to the next nearest open facility j in the first time window w' such that $\sum_{w=1}^{w'} |P^{jw}| < \sum_{w=1}^{w'} \chi_{nw}$. We then update the savings associated with closing the remaining open facilities which these population members have been reassigned to. Their savings values, in turn, depend on the effect that reassigning population members to their nearest alternative open facility

has on waiting times and treatment times. The procedure outlined in Algorithm 6 is utilized. Firstly, the selected facility j is closed by removing it from the set of open facilities (line 3). The next task is to assign the population members who were assigned to facility j to their next nearest open facility (line 8) and to keep track of the set of facilities whose will have more population members assigned to them (line 10). The set of population members assigned to facility j is cleared (line 13). The predicted waiting times for each facility and overall treatment time are updated using the machine learning algorithms (lines 16-19), these provide the reference values for updating the savings values associated with each of the remaining open facilities. We then update the savings values for each facility which has had population members reassigned to them due to the closure of facility j (the facility set μ), the savings values associated with the facilities not in this set ($O \setminus \{\mu\}$) will not have changed since its last update. To do this we need to calculate the number of population members that will be assigned to each other facility for each facility that can be closed next (lines 23-33). Then the machine learning algorithms are used to predict next possible waiting times and overall treatment times associated with each facility which can be closed next (lines 38-41). The deterministic and non-dynamic saving contributions are opening costs, assignment costs and travel times (lines 22 and 32 respectively). Once the savings list is updated a new facility can be selected for closure.

Algorithm 6 Close facility j and update savings of remaining open facilities. $updateSavings(j)$

Inputs: j (closing facility)
//Close facility j
$O \leftarrow O \setminus \{j\}$
//Initialize the set of facilities whose number of assigned population members will increase.
$\mu \leftarrow \emptyset$
for $i \in P^j$ **do**
  //Add the population member to their new facility $N(i)$ and to their new time window $L(i)$.
  $P^{N(i)L(i)} \leftarrow P^{N(i)L(i)} \cup \{i\}$
  // Update the set of facilities whose assigned number of population members will change.
  $\mu \leftarrow \mu \cup \{N(i)\}$
end for
//Clear the set of population members assigned to facility j.
$P^j \leftarrow \emptyset$
//Update the predicted waiting times for each remaining open facility $\tilde{t}_j^q$ and the overall treatment time $\tilde{T}^o$.
$\tilde{T}^o = 0$
for $k \in O$ **do**
  $\tilde{t}_k^q = ML_k^q \left(\sum_{w=1}^{|W|} |P^{kw}| \right)$
  $\tilde{T}^o = \max \left(\tilde{T}^o, ML_k^o \left(\sum_{w=1}^{|W|} |P^{kw}| \right) \right)$
end for
//Update the savings values for closing facilities in μ.
for $k \in \mu$ **do**
  $saving_k = (1-\psi)\, c_k$
  //Initialize storage of the new number of assigned population members of the remaining open facilities if facility k is closed.
  $\varepsilon_l = \sum_{w=1}^{|W|} |P^{lw}|, \ \forall l \in J$
  //Initialize the set of facilities whose number of assigned population members will increase if facility k is closed.
  $\nu \leftarrow \emptyset$
  //Add the number of population members assigned to facility k to the next nearest open facility in the first available time window.
  for $i \in P^k$ **do**
    $\varepsilon_{N(i)} = \varepsilon_{N(i)} + 1$
    $\nu \leftarrow \nu \cup \{N(i)\}$
    //Update savings value account for fixed costs and travel distance changes.
    $saving_k = saving_k + (1-\psi)\left(e_{ik} - e_{iN(i)}\right) + \psi\alpha\left(d_{ik} - d_{iN(i)}\right)$
  end for
  //Initialize new predicted overall treatment time if facility k is closed.
  $\hat{T}^o = \sum_{l \notin \nu \cup \{k\}} \max \left(\hat{T}^o, ML_l^o \left(\sum_{w=1}^{|W|} |P^{lw}| \right) \right)$
  //Account for savings due to waiting time changes due closing facility k using machine learning algorithm predictions.
  $saving_k = saving_k + \tilde{t}_k^q$
  for $l \in \nu$ **do**
    $saving_k = saving_k + \psi \left(\beta \left(\tilde{t}_l^q - ML_l^q \left(\sum_{w=1}^{|W|} \varepsilon_l \right) \right) \right)$
    $\hat{T}^o = \max \left(\hat{T}^o, ML_l^o \left(\sum_{w=1}^{|W|} \varepsilon_l \right) \right)$
  end for
  //Account for the predicted change in overall treatment time.
  $saving_k = saving_k + \psi\gamma\left(\tilde{T}^o - \hat{T}^o\right)$
end for
Output: $saving_k, \ \forall k \in O$

9 Computational Experiments

In Section 9.1, we consider the effects of different population arrival time distributions in conjunction with a range of time window durations. In Section 9.2, we consider how to allocate a simulation budget for generating input data for the machine learning module. In Section 9.3, we compare the proposed learnheuristic with exact, simulation-based and greedy algorithm alternatives for a range of problem types and sizes. Finally, in Section 9.4, we analyze the trade offs between solutions cost, travel times, waiting time and total treatment time.

9.1 Arrival Distribution, Variance and Time Window Duration

The results in this section are based on the case of 2000 population members arriving at a single facility which treats one person every minute. The results reported in Table 1 are each based on 10000 repeats of the simulation model outlined in Section 6. The experiments is repeated for all combinations of two different arrival time distributions (uniform random and triangle), three arrival time ranges (80,100 and 120) and three different values for time window duration (80, 100 and 120). Both arrival time distributions are symmetrical and the average arrival time of population members within their assigned time window is the middle of that time window. In cases where the arrival time range exceeds the time window duration, we model both early and late arrivals. For the case where the time window duration is 100 and the arrival time range is 80, the average arrival time will be 50 minutes into the time windows and all arrival will occur between 10 and 90 minutes into the time windows—which implies a gap between consecutive time windows which may be useful for clearing queues which remain at the end of time windows.

Table 1: The effect of arrival distributions, arrival time range variance on total waiting and overall treatment time.

		Average waiting time			Average total treatment time		
Arrival distribution	Arrival time range	Time window duration					
		80	100	120	80	100	120
Uniform	80	19064.0	27862.6	42784.5	2011.9	2015.6	2025.4
	100	25022.0	20541.6	27959.1	2015.2	2012.9	2031.2
	120	27401.5	26571.0	20959.2	2020.7	2016.2	2040.6
Triangle	80	31662.2	46311.7	60958.6	2057.6	2064.9	2073.7
	100	25035.5	37368.3	50508.8	2064.6	2070.7	2079.4
	120	20719.5	30200.0	41515.2	2074.4	2077.4	2093.3

Table 1 shows that a uniform random arrival pattern helps to reduce both waiting time and treatment time in all cases, this makes sense since the reduced rate of arrivals at the beginning of time windows, in the case of triangle distributed arrivals, leaves less time to treat the increased number of arrivals that occur in the remainder of the time window. The increased rate of arrivals that occurs in the middle of time window then leads to larger queues and hence larger waiting times. The next most obvious pattern, that is visible in Table 1, is that smaller time windows help to reduce overall treatment times. The reason for this is that the final time windows typically have very few population members assigned to them, but people may still arrive towards the end of the time window, as a result reducing the duration of time windows helps to improve the efficient use of the final time window. Table 1 reveals that, for the case of uniform random arrivals waiting times are reduced when the time window duration matches the arrival time range. When the arrival range exceeds the time window duration, people arrive before the beginning of the first time window and creates in initial queue that can propagate through the remainder of the process. When the arrival range is less than the time window duration people arrive in a greater concentration which increases the risk of queue formation. It is worth noting that: more people can be allocated to larger time windows. Although the average arrival rate is the same, larger time windows with proportionately more people assigned to them, have a greater potential for larger queues, and therefore waiting times and total treatment times. For the case of a triangle distribution smaller time windows reduce average waiting times in all

cases, which is because triangle distributed arrival patterns always induce a risk of queue formation in the middle of time windows, and smaller time windows means that fewer people are assigned

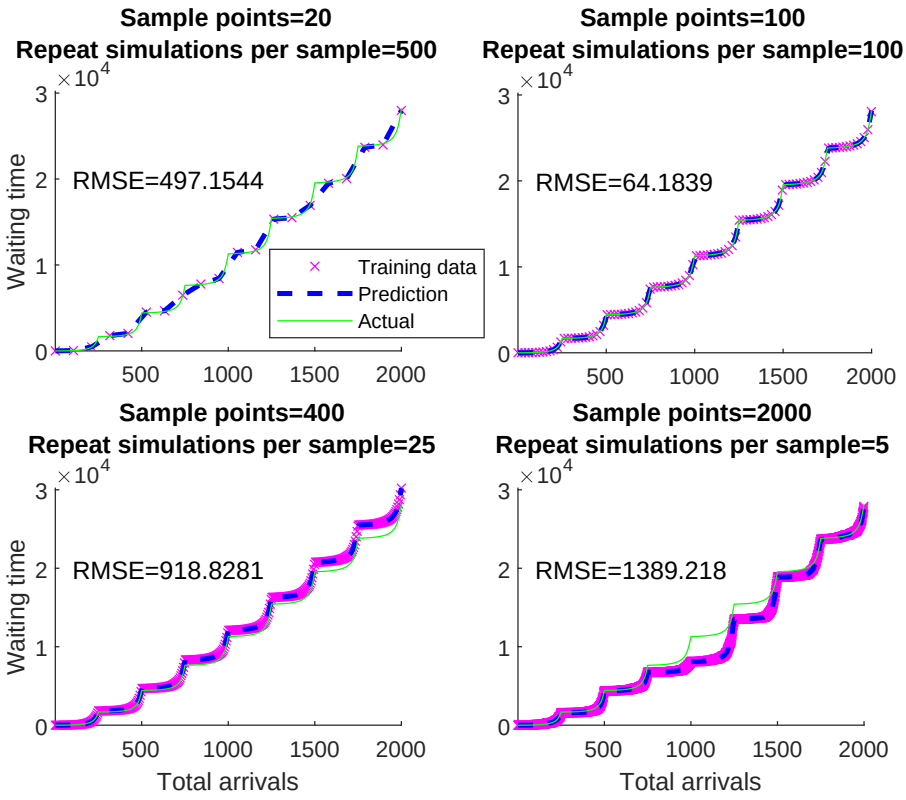

Figure 4: The effect of simulation budget utilization on waiting time prediction accuracy.

to each time window which reduces the average number of arrivals in the middle of each time window which reduces queue sizes as a result. In fact, reducing time windows should have the effect of making the overall arrival pattern tend to a uniform random arrival process, regardless of the arrival time distribution of the population. Small time windows corresponds to a flexible appointment approach.

9.2 Utilization of Simulation Budget for Generating Input Data for the ML Module

Given a budget of simulation repeats, we can choose to sample few scenarios each evaluated using a large number of repeat simulations, or sample a wide range of scenarios each evaluated using few repeat simulations. In the former case we have high accuracy estimates of few points, in the latter we have low accuracy estimates of many points. The ideal trade-off is to have quite accurate estimates of quite a lot of points. Figures 4 and 5 show how the sample size (H_j) effects prediction accuracy when there is a fixed simulation budget for evaluating those sample points. The RMSE values in those figures were calculated by comparing nearest neighbor with linear interpolation predictions with values calculated from 1000 repeat simulations for all possible number of arrivals between 1 and p_{max}. Figure 4 shows that, for the case of waiting times, a simulation budget of 10000, a maximum population size of 2000, average treatment time of 1 minute, times windows of 250 minutes with 250 treatments per time window, it was found that 100 sample points each evaluated using 100 repeat simulations results in the lowest root mean square error (RMSE). Fewer sample points, evaluated using more repeat simulations, fails to improve prediction accuracy because the linear interpolations between sampled points become more inaccurate. More sample points evaluated using fewer repeat simulations leads to inaccurate data, as shown by the increasing deviation of the training data and predictions from the actual waiting times. Figure 5 illustrates a similar effect for the treatment time predictions, where 400 sample points each evaluated using 25 repeat simulations leads to the lowest RMSE.

From figures 4 and 5, we can also conclude that treatment times are easier to predict than waiting times. Firstly, we see that we are able to obtain more accurate treatment time predictions based on data generated using fewer repeat simulations per sample point. Secondly, this makes sense from the perspective that arrival patterns with different total waiting times can lead to the same overall treatment time. For instance, a uniform arrival pattern and an arrival pattern where all population members arriving at the beginning of their assigned time window can both have equal total treatment time but vastly different total waiting times.

Figures 4 and 5 also reveal non-linearity and discontinuity in total waiting and treatment times as the number of assigned population members is increased. In this case, the discontinuities occur at intervals of 250 minutes, which is the duration of a time window. The jumps are sharper for the case of total treatment times and correspond to the need to allocate very few people to a new time window because the earlier ones are full. For the case of waiting times, these jumps are accompanied by a superlinear increase, which occurs as the number of people assigned to the last time window approaches its maximum value. This reveals that waiting times can be greatly reduced by under-subscribing time windows. Furthermore, a larger scale superlinear increase in waiting time occurs due to a cumulative effect of time windows overflowing into the next one. This means that breaks in between consecutive time windows can be beneficial for reducing total waiting times.

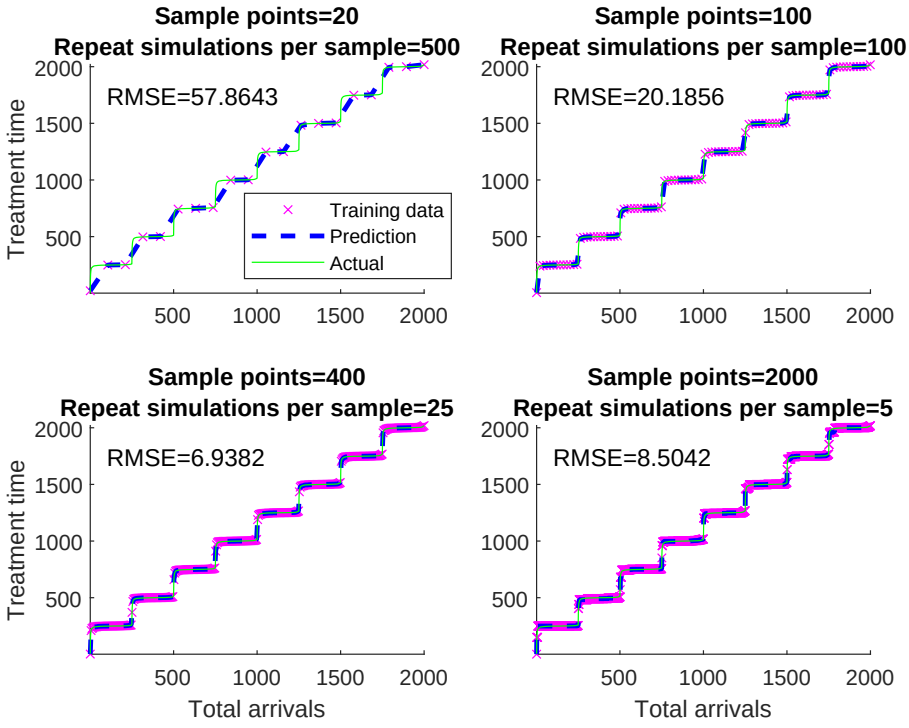

Figure 5: The effect of simulation budget utilisation on treatment time prediction accuracy.

9.3 Comparison of Algorithms over a Range of Problem Sizes

This section compares the proposed learnheuristic (LH) with the exact formulation (EXACT) solved using the commercial solver CPLEX, a simulation equivalent of the proposed method (SH) and a greedy heuristic implementation based on the proposed learnheuristic (GLH). The termination criteria for the EXACT approach is to terminate upon the first of two events, 1) proven optimal solution found; or 2) solution time reaches 7200 seconds. In particular, SH is exactly the same as LH except that every time LH uses a machine algorithm prediction for evaluating decision costs, SH uses the simulation. SH will allow us to demonstrate what is gained by using LH instead of a simulation optimization based approach. LH and SH both perform 500 iterations with 5 applications

of the biased randomized constructive heuristic in each learnheuristic iteration, λ is selected randomly in $[0,1]$ at the beginning of each learnheuristic iteration. GLH is exactly the same as LH except that only one LH iteration performed and the parameter λ is set to 1 so that the constructive heuristic behaves as a greedy algorithm (**?**).

Table 2: Comparison of algorithms for a variety of problem sizes.

$\|S\|$	$\|J\|$	$\|P\|$	$\|m_{jw}\|$	Method	Objective value	Costs	Travel time	Waiting time	Treatment time	Solution time s (log_{10})
1	1	20	20	EXACT	34.93	40.85	7.23	30.52	23.74	-1.23
				SH	34.93	40.85	7.23	30.52	23.74	-1.98
				LH	34.93	40.85	7.23	30.52	23.74	-2.06
				GLH	34.93	40.85	7.23	30.52	23.74	-2.55
2	2	40	20	EXACT	53.06	55.35	15.96	75.41	48.47	1.49
				SH	53.06	55.35	15.96	75.41	48.47	-1.44
				LH	53.06	55.35	15.96	75.41	48.47	-1.87
				GLH	53.06	55.35	15.96	75.41	48.47	-2.13
3	4	80	40	EXACT	95.57	87.9	32.09	211.15	90.58	3.8
				SH	95.93	82.98	33.37	231.89	106.8	-0.56
				LH	95.93	82.98	33.37	231.89	106.8	-1.36
				GLH	96.16	85.52	31.77	219.28	105.52	-1.84
4	8	160	40	EXACT	-	-	-	-	-	-
				SH	175.6	168.35	50.66	366.17	154.27	0.31
				LH	175.6	168.35	50.66	366.17	154.27	-0.74
				GLH	175.88	168.25	48.99	367.5	158.29	-1.4
5	16	320	80	EXACT	-	-	-	-	-	-
				SH	325.36	337.93	83.86	580.83	222.64	0.84
				LH	325.26	337.93	83.25	580.42	222.67	-0.23
				GLH	328.74	408.74	71.64	248.99	114.64	-0.75
10	32	640	80	EXACT	-	-	-	-	-	-
				SH	639.13	629.1	137.69	1496.34	373.55	1.79
				LH	639.6	630.55	139.53	1509.14	352.3	0.33
				GLH	645.16	805.32	110.48	582.41	151.11	-0.1
20	64	1280	160	EXACT	-	-	-	-	-	-
				SH	1035.3	1299.77	201.67	840.84	267.53	2.84
				LH	1035.14	1299.24	201.61	858.82	250.53	1
				GLH	1042.57	1324.97	200.99	755.13	250.74	0.6
30	128	2560	160	EXACT	-	-	-	-	-	-
				SH	2046.76	2602.77	331.49	1674.6	373.48	3.64
				LH	2046.53	2603.26	334.56	1666.96	372.63	1.76
				GLH	2057.2	2628.43	329.62	1592.15	383.08	1.41
40	256	5120	320	EXACT	-	-	-	-	-	-
				SH	-	-	-	-	-	-
				LH	2884.76	3598.96	381.75	2606.57	676	2.6
				GLH	2897.07	3600.46	386.92	2708.8	681.23	2.07
50	512	10240	320	EXACT	-	-	-	-	-	-
				SH	-	-	-	-	-	-
				LH	5745.66	7208.54	567.54	5347.34	1141.28	3.39
				GLH	5760.43	7229.63	564.99	5311.24	1177.16	2.72
100	1024	20480	640	EXACT	-	-	-	-	-	-
				SH	-	-	-	-	-	-
				LH	-	-	-	-	-	-
				GLH	9957.42	12773.4	988.61	7425.32	1796.76	3.37

For each problem size a range of different problem types are considered. The types of instance are varied according to the spatial distribution of nodes, two cases are considered: *(i)* x and y coordinates of each node are selected according to a uniform random distribution in $[0,1]$; and *(ii)* the x and y coordinates of nodes are generated

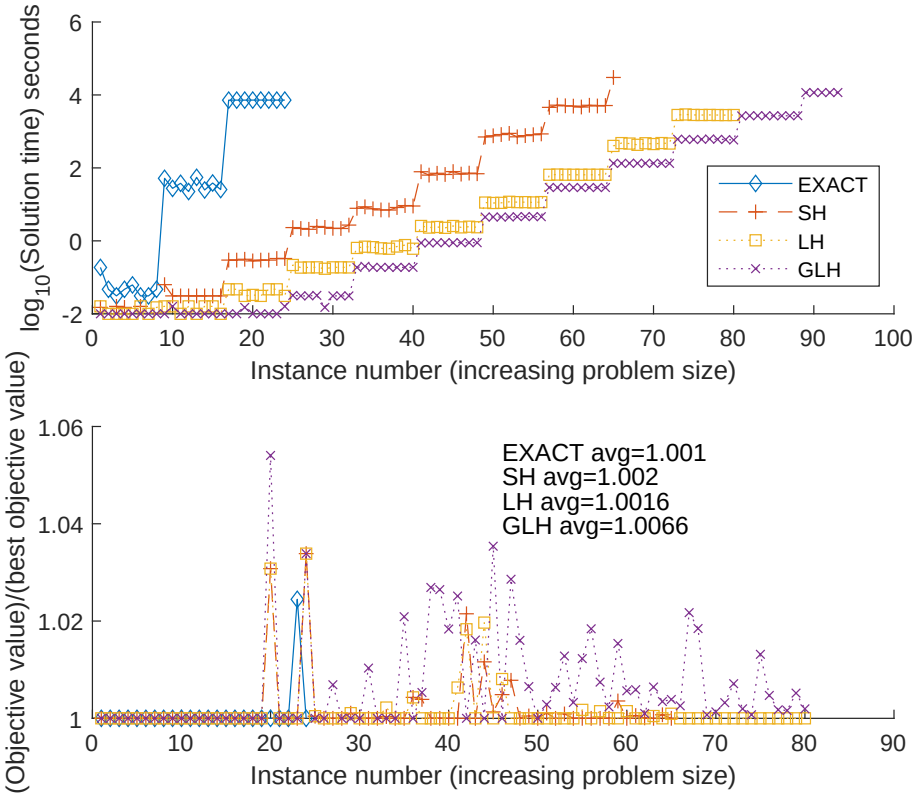

Figure 6: Solutions times and objective values relative to the best known for all solution methodologies for all sizes and types.

using polar coordinates, firstly an angle is selected $\theta \in [0,360)$ and a radius $r \in [0,0.5]$, then $x = rcos(\theta)$ and $y = rsin(\theta)$. In this approach nodes are distributed in a circle with increased density in the center, resembling a city-like arrangement. The instance types for each problem size are also varied according constant (30) or random (uniform random $[0,30]$) facility costs and constant (1) or random (uniform random $[0,1]$) treatment costs per population member, i.e., 4 combinations, making 8 instances for each problem size. The details regarding the problem sizes considered are given in Table 2.

The results in Table 2 report the objective values, solution costs, travel time, waiting time, overall treatment time and solution time for each solution methodology on average over the 8 instances for each problem size. Solution times are given as log_{10}, which gives the number of zeros/order of magnitude of solution time in seconds, i.e., 2=100 seconds and 3=1000 seconds. These statistics are calculated by testing the final solutions in 500 repeat simulations starting from a different random seed to that used during the optimization process. Figure 6 plots the objective values, relative to the best achieved, and solution times for all solution methodologies, all problem sizes and all problem types. Table 2 and Figure 6 firstly show that the EXACT approach was only able to solve instances in the three smallest problem sizes before out of memory errors occurred. For these instances the heuristic methodologies each found solution of equal or similar quality and in a fraction of the time required by the EXACT approach. For the next five sets of increasingly sized problems SH, LH and GLH were able to provide solutions, with SH and LH providing solutions of equal or very similar quality. GLH provided notable lower quality solutions, which highlights the benefit of biased randomization. For larger problem sizes solution times for SH became an issue due to its excessive use of the simulation module, which highlights the benefit of the LH, that of allowing us to tackle even larger problems. Regarding the significance of the differences between the 4 algorithms, only GLH gave solutions with noticeably different objective values to those of EXACT, SH and LH. Based on independent sample t-tests, the difference between GLH and LH was significant at the 0.05 α level in

instances involving populations of 320, 640, 1280 and 2560. In the other instances GLH was able to find solutions of comparable quality to LH such that the reported differences have a greater than 5% chance of being attributable to chance.

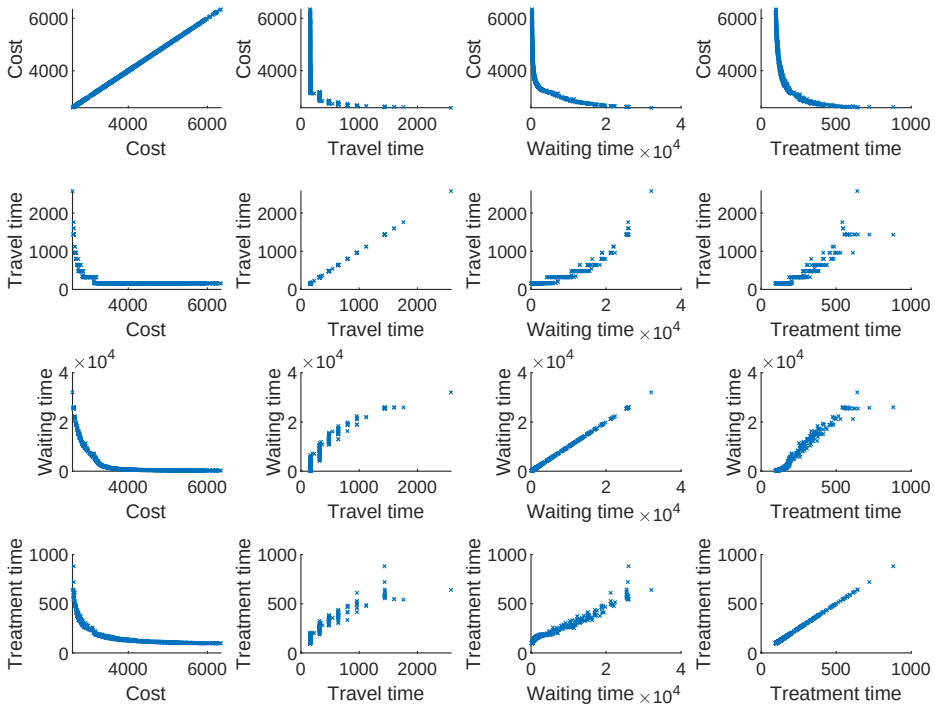

Figure 7: Matrix scatter plot of the costs, travel times, waiting times and overall treatment times associated with the efficient solutions.

9.4 The Trade Off Between Costs, Travel Time, Waiting Time and Completion Time

The experiments in this section are based on an example problem instance involving a population size of 2560, a maximum number of facilities of 128, a time window size of 320, treatment times of one minute, population members arrive at their assigned facility according to a uniform random distribution during their assigned time window, the nodes distributed according to the polar coordinate approach described in Section 9.3, treatment costs and facility opening costs are fixed at 1 and 30 respectively.

In order to explore the trade offs between the considered objectives 5000 LH iterations were performed, with the other parameters set to the same values as those specified in Section 9.3, and the set of non-dominated solutions were collected. In each LH iteration the weights given to each objective $(\psi, \alpha, \beta, \gamma)$ were randomized according to a uniform random distribution, $\psi \in [0,1]$, α, β, γ were similarly sampled before being divided by their sum, such that their sum was 1. A solution is dominated if there exists another solution that has better or equal characteristics across all objectives with at least one of them being better. The set of non-dominated is the composite of the set of dominated solutions and defines an efficient frontier of solutions. Figure 7 plots this efficient set of solutions using a matrix of scatter plots, with rows and columns corresponding to each objective, this allows us to judge the nature of the trade offs between each pair of objectives.

Figure 7 shows that the cost is a conflicting objective with travel time, waiting time and treatment time. The reason for this is that reducing travel time, waiting time and treatment time requires the opening of more facilities which increases costs. Figure 7 also shows that there are diminishing returns for each extra facility that is opened, as costs approach 4000 the reductions in travel time, waiting time and treatment time approach zero. Travel time, waiting time and treatment time are in general non-conflicting objectives. Waiting time and treatment time only

become competing objectives when optimizing the approach for controlling the arrival process, i.e., large queues ensure high server utilization, which reduces overall treatment time at the expense of increased waiting times.

10 Conclusions

This work considers an integrated facility location and queue optimization problem, and outlines a scalable optimization approach for determining the set of facilities to open and which facilities population members can be assigned to. The problem was formulated as a mixed integer program which could be solved for small instances. In order to solve large instances of this stochastic problem, a learheuristic algorithm was proposed. It integrates a biased randomization algorithm with simulation and machine learning components. The simulation model was used provide training data for the machine learning algorithm and for evaluating promising solutions that were identified by the learnheuristic algorithm. The machine learning algorithm was used to avoid excessive use of the simulation model when updating the solution dependent decision costs within the biased randomized constructive algorithm.

Population members were assigned to facilities as well as time windows, in a flexible appointment fashion. Experimental results showed that the choice of time window duration should be chosen as a function of the arrival time distribution of the population, and that, in general, smaller time windows help to reduce average waiting times as well as total treatment times regardless of the actual arrival time distribution of the population. Also, the way in which a budget of simulation runs is used to generate input data for the machine learning algorithm of the learnheuristic is an important design decision. In particular, we are faced with finding a balanced trade off between a large amount of inaccurate input data and a small amount of accurate input data.

In comparison with exact solutions, which were only available for small relatively trivial problem instances, the proposed heuristic approaches provided solutions of equal or similar quality in a fraction of the time required by the exact approach. The solution times of simulation-only implementation of the learnheuristic became inordinately large as the problems sizes were increased. The learnheuristic approach was between 1 and 2 orders of magnitude faster than the simulation-only approach, while at the same time providing solution of equal or very similar quality to the the simulation-only approach.

Experimental results revealed the nature of the trade offs between cost, travel time, waiting time and treatment time. In general, cost was a conflicting objective with travel time, waiting time and treatment time, namely because reducing travel, waiting and treatment times requires the opening of more facilities, which increase costs. The results also indicated a case of diminishing returns from increasing the number of facilities opened.

Acknowledgements

This work has been partially supported by the Erasmus+ programme (2019-I-ES01-KA103-062602) and by the Spanish Ministry of Science, Innovation, and Universities (RED2018-102642-T).

REFERENCES

Ageron, B., Benzidia, S., and Bourlakis, M. (2018). Healthcare logistics and supply chain–issues and future challenges. In *Supply Chain Forum: An International Journal*, volume 19, pages 1–3. Taylor & Francis.

Ahmadi-Javid, A., Seyedi, P., and Syam, S. S. (2017). A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263.

Balinski, M. L. (1964a). On finding integer solutions to linear programs. Technical report, Mathematica, Princeton, NJ, USA.

Balinski, M. L. (1964b). On finding integer solutions to linear programs. Technical report, MATHEMATICA PRINCETON NJ.

Bayliss, C., Juan, A. A., Currie, C. S., and Panadero, J. (2020). A learnheuristic approach for the team orienteering problem with aerial drone motion constraints. *Applied Soft Computing Journal*, 92:106280.

Calvet, L., de Armas, J., Masip, D., and Juan, A. A. (2017). Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open Mathematics*, 15(1):261–280.

Castaneda, J., Ghorbani, E., Ammouriova, M., Panadero, J., and Juan, A. A. (2022). Optimizing transport logistics under uncertainty with simheuristics: Concepts, review and trends. *Logistics*, 6(3):42.

Chaleshtori, A. E., Jahani, H., and Aghaie, A. (2020). Bi-objective optimization approach to a multi-layer location–allocation problem with jockeying. *Computers & Industrial Engineering*, 149:106740.

Correia, I. and Saldanha-da Gama, F. (2019). Facility location under uncertainty. In *Location science*, pages 185–213. Springer.

Daskin, M. S. and Dean, L. K. (2005). Location of health care facilities. In *Operations research and health care*, pages 43–76. Springer.

de Armas, J., Juan, A. A., Marquès, J. M., and Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176.

do C. Martins, L., Hirsch, P., and Juan, A. A. (2021). Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *International Transactions in Operational Research*, 28(1):201–221.

Efroymson, M. and Ray, T. (1966). A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368.

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009.

Estrada-Moreno, A., Ferrer, A., Juan, A. A., Bagirov, A., and Panadero, J. (2020). A biased-randomised algorithm for the capacitated facility location problem with soft constraints. *Journal of the Operational Research Society*, 71(11):1799–1815.

Fomundam, S. and Herrmann, J. W. (2007). A survey of queuing theory applications in healthcare. Technical report.

Grasas, A., Juan, A. A., and Lourenço, H. R. (2016). Simils: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77.

Gu, W., Wang, X., and McGregor, S. E. (2010). Optimization of preventive health care facility locations. *International journal of health geographics*, 9(1):17.

Hakli, H. and Ortacay, Z. (2019). An improved scatter search algorithm for the uncapacitated facility location problem. *Computers & Industrial Engineering*, 135:855–867.

Jacobson, S. H., Hall, S. N., and Swisher, J. R. (2006). Discrete-event simulation of health care systems. In *Patient flow: Reducing delay in healthcare delivery*, pages 211–252. Springer.

Jahre, M., Persson, G., Kovács, G., and Spens, K. M. (2007). Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management*.

Jun, J., Jacobson, S. H., and Swisher, J. R. (1999). Application of discrete-event simulation in health care clinics: A survey. *Journal of the operational research society*, 50(2):109–123.

Körkel, M. (1989). On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research*, 39(2):157–173.

Lai, M.-C., Sohn, H.-s., Tseng, T.-L. B., and Chiang, C. (2010). A hybrid algorithm for capacitated plant location problem. *Expert Systems with Applications*, 37(12):8599–8605.

Lakshmi, C. and Iyer, S. A. (2013). Application of queueing theory in health care: A literature review. *Operations research for health care*, 2(1-2):25–39.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2019). Iterated local search: Framework and applications. In *Handbook of metaheuristics*, pages 129–168. Springer.

Martinez, G., Huschka, T., Sir, M., and Pasupathy, K. (2016). A coordinated scheduling policy to improve patient access to surgical services. In *2016 Winter Simulation Conference (WSC)*, pages 2041–2052. IEEE.

Martins, L. d. C., Tarchi, D., Juan, A. A., and Fusco, A. (2022). Agile optimization for a real-time facility location problem in internet of vehicles networks. *Networks*, 79(4):501–514.

Monks, T. and Meskarian, R. (2017). Using simulation to help hospitals reduce emergency department waiting times: Examples and impact. In *2017 Winter Simulation Conference (WSC)*, pages 2752–2763. IEEE.

Moons, K., Waeyenbergh, G., and Pintelon, L. (2019). Measuring the logistics performance of internal hospital supply chains–a literature study. *Omega*, 82:205–217.

Nuñez-Perez, N., Ortíz-Barrios, M., McClean, S., Salas-Navarro, K., Jimenez-Delgado, G., and Castillo-Zea, A. (2017). Discrete-event simulation to reduce waiting time in accident and emergency departments: a case study in a district general clinic. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 352–363. Springer.

Oksuz, M. K. and Satoglu, S. I. (2020). A two-stage stochastic model for location planning of temporary medical centers for disaster response. *International Journal of Disaster Risk Reduction*, 44:101426.

Quintero-Araujo, C. L., Guimarans, D., and Juan, A. A. (2021). A simheuristic algorithm for the capacitated location routing problem with stochastic demands. *Journal of Simulation*, 15(3):217–234.

Reese, H. D., Anandhan, V., Pérez, E., and Novoa, C. (2017). Improving patient waiting time at a pure walk-in clinic. In *2017 Winter Simulation Conference (WSC)*, pages 2764–2773. IEEE.

Rohleder, T. R., Lewkonia, P., Bischak, D. P., Duffy, P., and Hendijani, R. (2011). Using simulation modeling to improve patient flow at an outpatient orthopedic clinic. *Health care management science*, 14(2):135–145.

Silva, A., Aloise, D., Coelho, L. C., and Rocha, C. (2021). Heuristics for the dynamic facility location problem with modular capacities. *European journal of operational research*, 290(2):435–452.

Silva, F. and Serra, D. (2008). Locating emergency services with different priorities: the priority queuing covering location problem. *Journal of the Operational Research Society*, 59(9):1229–1238.

Tavakkoli-Moghaddam, R., Vazifeh-Noshafagh, S., Taleizadeh, A. A., Hajipour, V., and Mahmoudi, A. (2017). Pricing and location decisions in multi-objective facility location problem with m/m/m/k queuing systems. *Engineering Optimization*, 49(1):136–160.

Uno, T., Katagiri, H., and Kato, K. (2010). A facility location for fuzzy random demands in a competitive environment. *IAENG International Journal of Applied Mathematics*, 40(3).

Verma, A., Verma, R., and Mahanti, N. (2010). A new approach to fuzzy uncapacitated facility location problem. *International Journal of Soft Computing*, 5(3):149–154.

Verter, V. and Lapierre, S. D. (2002). Location of preventive health care facilities. *Annals of Operations Research*, 110(1-4):123–132.

Wang, L., Zhang, Z., Wu, C., Xu, D., and Zhang, X. (2021). Approximation algorithms for the dynamic k-level facility location problems. *Theoretical Computer Science*, 853:43–56.

Witten, I. H., Frank, E., and A.Hall, M. (2011). *Data mining: Practical machine learning tools and techniques*. Elsevier, Burlington, USA.

Wu, T., Huang, L., Liang, Z., Zhang, X., and Zhang, C. (2022). A supervised learning-driven heuristic for solving the facility location and production planning problem. *European Journal of Operational Research*, 301(2):785–796.

Zamani, S., Arkat, J., and Niaki, S. T. A. (2022). Service interruption and customer withdrawal in the congested facility location problem. *Transportation Research Part E: Logistics and Transportation Review*, 165:102866.

Zambrano, F., Concha, P., Ramis, F., Neriz, L., Bull, M., Veloz, P., and Carvajal, J. (2016). Improving patient access to a public hospital complex using agent simulation. In *2016 Winter Simulation Conference (WSC)*, pages 1277–1288. IEEE.

Zhang, Y., Berman, O., and Verter, V. (2009). Incorporating congestion in preventive healthcare facility network design. *European Journal of Operational Research*, 198(3):922–935.