European Journal of Operational Research 319 (2024) 31–49

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Discrete optimization

An exact algorithm for the multi-trip vehicle routing problem with time windows and multi-skilled manpower

Nan Huang [a], Hu Qin [a], Yuquan Du [b], Li Wang [c,*]

[a] School of Management, Huazhong University of Science and Technology, Wuhan, China
[b] La Trobe Business School, La Trobe University, Melbourne, VIC 3086, Australia
[c] School of Economics and Management, Hebei University of Technology, Tianjin, China

ARTICLE INFO

Keywords:
Multi-trip
Multi-skilled
Manpower planning
Branch-and-price-and-cut
Two-phase column generation

ABSTRACT

Motivated by the challenges of non-emergency patient transportation services in the healthcare industry, this study investigated a multi-trip vehicle routing problem incorporating multi-skilled manpower with downgrading. We aimed to find an optimal plan for vehicle routing and multi-skilled manpower scheduling in tandem with the objective of minimizing the total cost, including travel and staff costs, without violating time windows and lunch break constraints. To address this, two mathematical models were formulated: an arc-flow model and a trip-based set-covering model. In addition, a branch-and-price-and-cut algorithm, based on the set-covering model, was proposed to solve practical-scale instances. To determine the feasibility of the integer solutions, we introduce a feasibility check model. To address the multi-trip characteristics of the proposed problem, a novel two-phase column generation algorithm was introduced to solve the subproblem. This approach differs from traditional one-phase labeling algorithms and involves a tailored labeling algorithm for obtaining non-dominated labels in the first phase and a strategy to identify the trip with the minimum reduced cost for each label in the second phase. Furthermore, novel and efficient staff-based inequalities were developed by improving the k-path inequalities. Extensive numerical experiments were conducted to demonstrate the solution performance of the proposed algorithm and reveal managerial insights for non-emergency ambulance operations. The results demonstrate that our algorithm can successfully solve instances with up to 50 patients to optimality within two hours. Moreover, we demonstrated the value of jointly optimizing vehicle routing and staff planning, which can result in significant cost savings of up to 19.4%.

1. Introduction

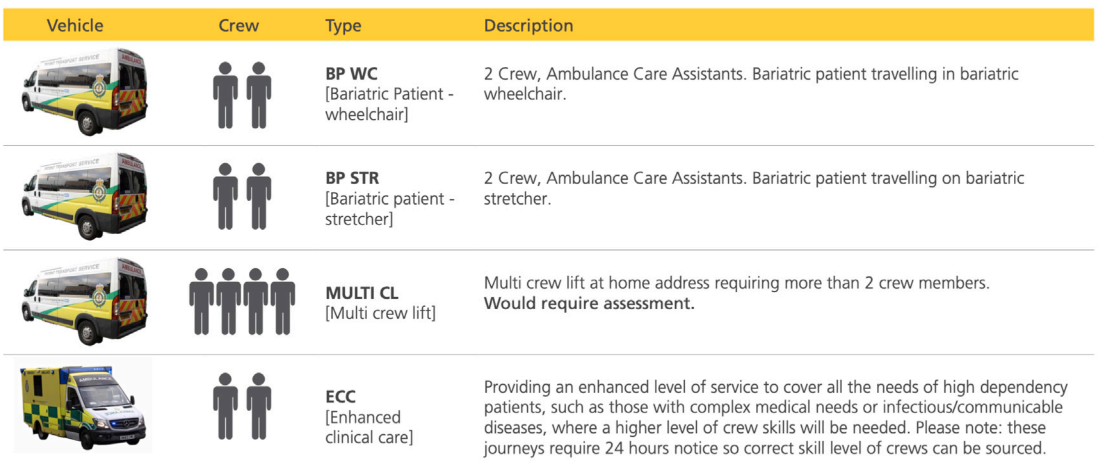
According to a survey by the National Health Service in the UK, approximately 90% of people believe that the journey of patients toward treatment represents a significant contribution to their perception of the healthcare service level (England NHS, 2021). However, most patients, particularly those who cannot use other transportation types owing to medical conditions or severe mobility issues, suffer from inefficient patient transport. For instance, approximately 47% of patients were late for a hospital appointment owing to transport issues, even though the patients were usually required to be ready for pick-up at least two hours before the given collection time. The aging population in many societies exacerbate the adverse effects of inefficient patient transport. To address this issue, non-emergency patient transport service (NEPTS) providers have been established in many areas, such as the UK, America, Canada, Singapore, Australia, and Hong Kong. NEPTS is engaged in carrying out the appropriate and efficient transport of patients within a defined time by managing a fleet of equipped liveried ambulances and a group of non-emergency ambulance staff. Correspondingly, service providers face many decision-making challenges such as vehicle routing, route scheduling, and manpower allocation. Therefore, it is important for NEPTS providers to develop efficient and cost-effective non-emergency patient transportation solutions to provide high-quality services while minimizing costs.

Motivated by the challenges arising from non-emergency patient transport services, this study investigated a multi-trip vehicle routing problem with time windows and multi-skilled manpower (MTVRPTW-MSM). Specifically, given a fleet of ambulances and operative staff, the service provider arranges the transport plan for patients from their residences to the hospital according to the requests registered in advance. Each ambulance is equipped with a fixed number of adaptive seating spots that can be converted into beds or wheelchairs when necessary (Fig. 1). When a patient requires a wheelchair, two seating spots are required, and three seating spots are required for a bed. Each

* Corresponding author.
E-mail addresses: nan_huang@hotmail.com (N. Huang), tigerqin@hust.edu.cn (H. Qin), bill.du@latrobe.edu.au (Y. Du), liliw0227@gmail.com (L. Wang).

https://doi.org/10.1016/j.ejor.2024.06.025
Received 15 May 2023; Accepted 17 June 2024
Available online 20 June 2024

Fig. 1. An ambulance with adaptive seats. Zhang et al. (2017).



Vehicle	Crew	Type	Description
		BP WC [Bariatric Patient - wheelchair]	2 Crew, Ambulance Care Assistants. Bariatric patient travelling in bariatric wheelchair.
		BP STR [Bariatric patient - stretcher]	2 Crew, Ambulance Care Assistants. Bariatric patient travelling on bariatric stretcher.
		MULTI CL [Multi crew lift]	Multi crew lift at home address requiring more than 2 crew members. **Would require assessment.**
		ECC [Enhanced clinical care]	Providing an enhanced level of service to cover all the needs of high dependency patients, such as those with complex medical needs or infectious/communicable diseases, where a higher level of crew skills will be needed. Please note: these journeys require 24 hours notice so correct skill level of crews can be sourced.

Fig. 2. Some ambulance service types provided by a service provider SCAS (2017).

request for a patient includes the pick-up location and time window in which the patient is willing to be picked up. For patients who cannot complete the treatment process independently, transport arrangements should accommodate their escorts or caregivers as well. In such cases, a patient request may need more than one seat in the ambulance. Moreover, some patients require special skills or support from ambulance staff. In accordance with the different medical conditions and mobility situations of patients, ambulances are manned by operatives with multiple levels of skills and qualifications. In this context, it is natural for the service provider to allow skill *downgrading*, in which operatives have higher qualifications and skill levels than those required by some customers/patients (Bard, 2004; Dall'Olio & Kolisch, 2023; Khodabandeh et al., 2021). Fig. 2 illustrates the equipment and staffing standards for certain service types provided by the *South Central Ambulance Service*, an NEPTS provider in the UK (SCAS, 2017).

Because of the limited capacity of each ambulance and the limited number of ambulances, an ambulance may cover multiple trips during working hours per day. An ambulance starts each trip from the hospital with staff, picks up several patients and their escorts or caregivers (if any), and transports them back to the hospital. The time window requirements for each patient should be satisfied for each trip. Each ambulance operative (including the driver) is supposed to have a lunch break at the hospital during a given interval. The ambulance load, including operatives and their customers (patients with their escorts), should be within the passenger-carrying capacity. The total number of occupied ambulance operatives at any time during the planning horizon should not exceed the staffing limit, that is, the total number of operatives available to the service provider. The planning objective is to minimize the total cost, which consists of the travel cost of the ambulances and the costs associated with staff payments, by simultaneously determining the optimal ambulance routes and manpower plans.

MTVRPTW-MSM extends the multi-trip vehicle routing problem with time windows (MTVRPTW) by incorporating multiskilled manpower planning. The MTVRPTW is a well-known NP-hard combinatorial optimization problem. Therefore, the MTVRPTW-MSM is also an NP-hard problem but with additional complexity. The MTVRPTW has attracted increasing attention in recent years. Various heuristic and exact solution methods have been developed to solve this. However, to the best of our knowledge, no existing studies have addressed the MTVRPTW by incorporating additional manpower planning requirements, which is practically relevant and significant for non-emergency patient transport services. From a theoretical perspective, the main challenge of the MTVRPTW-MSM is jointly optimizing trip routing, trip scheduling, and manpower planning.

The contributions of this study are summarized as follows:

First, we introduce a realistic and practical variant of the VRP, that simultaneously considers several constraints associated with multiple trips, time windows, multi-skilled manpower, and lunch breaks.

Second, from a methodological perspective, we develop two mathematical formulations for this innovative research problem: an arc-flow model and a trip-based set-covering model.

Third, to solve practical-scale instances of this problem, we propose an effective branch-and-price-and-cut (BPC) algorithm based on the set-covering model. We summarize the innovations of the proposed algorithm as follows:

- Regarding the characteristic of the multi-trip of the proposed problem, in contrast to the traditional one-phase labeling algorithm, we propose a novel two-phase column generation (CG) algorithm to solve the sub-problem. In the first phase, we propose a tailored labeling algorithm to obtain non-dominated labels. Each label corresponds to numerous trips in terms of a considerable number of possible departure times. In the second phase, we propose a well-designed strategy to identify a trip with the minimum reduced cost for each label.
- To avoid complicating the labeling algorithm with lunch break constraints, we first ignore the impact of lunch breaks on the sub-problem in the labeling algorithm and then deal with the lunch break constraints by searching for trips with minimum reduced cost.
- To verify whether the obtained integer solutions are feasible, we propose a feasibility check model to quickly judge the feasibility of the solutions. We also present resolution strategies for different infeasible solutions, namely halving the time intervals for case 3.5 and adding feasible cuts for case 1.
- Based on the proposed models, we originally propose staff-based inequalities which can be considered as an improvement over traditional k-path inequalities, by considering the additional seats occupied by staff according to the requests of patients on a trip. These inequalities can provide significant enhancements to the model compared to the traditional k-path inequalities.

Finally, based on Solomon's instances, extensive numerical experiments are conducted to verify the solution performance of our BPC algorithm and reveal managerial insights through sensitivity analyses of key parameters. Results show that our algorithm can successfully solve instances with up to 50 patient transportation requests, four vehicles (ambulances), and 10 staff members (ambulance operatives) to optimality within two hours. In addition, we demonstrate the value of jointly optimizing vehicle routing and staff planning, which can result in significant cost savings of up to 20.3%. The comprehensive experimental results can serve as a baseline for future research on this work and other related problems.

The remainder of this paper is organized as follows: Section 2 reviews relevant studies on MTVRPTW and the integrated optimization of vehicle routing and manpower planning. Section 3 formally defines the research problem and presents two mathematical formulations. Section 4 introduces the CG algorithm for solving the linear programming relaxation problem in the branch-and-bound (B&B) tree. In Section 5, we present the proposed BPC algorithm. Section 6 reports the results of the numerical experiments and discusses their managerial implications. Finally, Section 7 concludes the paper.

2. Literature review

To the best of our knowledge, we have taken the initiative to address the MTVRPTW with multiskilled manpower requirements. In this section, we first review the literature on multitrip vehicle routing problems with time windows, which are closely related to our study. Moreover, studies on the integrated optimization of routing and manpower planning are relevant. Therefore, we review existing studies on four types of research problems that combine the characteristics of both the vehicle routing problem (VRP) and manpower planning: VRP with multiple deliverymen, manpower routing with synchronization constraints, aircraft routing with crew pairing, and vehicle routing and manpower allocation problems in the context of patient transport.

2.1. Studies on MTVRPTW

Over the past decades, exact algorithms for the MTVRPTW have attracted increasing attention. Azi et al. (2007) introduced a single-vehicle version of MTVRPTW for perishable product transport. They proposed a two-phase exact algorithm in which all feasible, non-dominated trips are generated in the first phase, and some of them are connected to form feasible routes in the second phase. Azi et al. (2010) developed a set-partitioning model for MTVRPTW with multiple vehicles and proposed a branch-and-price solution algorithm. Macedo et al. (2011) and Hernandez et al. (2014) investigated the same problem and further improve the solution method. Some studies have addressed the MTVRPTW without a trip-duration constraint, which is often more computationally complicated. Hernandez et al. (2016) propose two set-covering formulations: one is route-based, and the other is trip-based. These models were solved using branch-and-price algorithms. Their experimental results indicate that the trip-based branch-and-price framework is more computationally efficient. Paradiso et al. (2020) constructed a structure-based formulation, introduced two mathematical programming relaxations to derive the lower bounds, and devised a BPC algorithm to solve the model. Yang (2022) improved the algorithm to solve instances with more customers. Huang et al. (2021) and Huang et al. (2024) introduced the multi-trip problem with queuing and proposed discrete and continuous BPC algorithms, respectively.

The research gap between these existing studies on the MTVRPTW and the non-emergency patient transport problem considered in this study lies in the absence of planning for the manpower involved, that is, the allocation and scheduling of operatives working with ambulances.

2.2. Studies on integrated optimization of routing and manpower planning

VRP with time windows and multiple deliverymen (VRPTWMD)

The VRPTWMD was first introduced by Pureza et al. (2012) for the distribution of beverages and tobacco in urban areas. In the VRPTWMD, the total service time at demand sites along a route is significantly longer than the total traveling time of the vehicle, which often results in a situation in which the crew drivers cannot complete their deliveries of all orders along a route within their regular working hours. If the maximum routing time of a vehicle along a route is beyond the crew drivers' working hours, penalties are applied in the form of overtime pay or expensive fines from the government. Therefore, extra deliverymen may need to be deployed on a vehicle route to assist delivery services at demand sites to control or compress the total service time, thereby reducing the maximum routing time of the vehicle. This represents a unique decision dimension of the VRPTWMD because the total service time over a route is dependent on the number of deliverymen assigned to it. Pureza et al. (2012) developed an arc flow formulation for VRPTWMD and designed two heuristic methods including ant colony optimization and tabu search. Alvarez and Munari (2017) investigated the same problem and designed a hybrid approach that combines a BPC algorithm with two meta heuristics. Munari and Morabito (2018) developed a set-partitioning formulation and proposed a BPC algorithm for the VRPTWMD, which is regarded as the first exact solution algorithm for the VRPTWMD. A few studies have focused on the possible uncertainties in VRPTWMD, such as uncertain demands (De La Vega et al., 2019), uncertain travel times, and service times (De La Vega et al., 2020). These studies employed a robust optimization approach with budgeted uncertainty sets and developed heuristic or exact solution algorithms.

These studies on the VRPTWMD have a large research gap in addressing the nonemergency patient transport problem considered in this study. This can be explained as follows. First, in the decision dimension of vehicle routing, non-emergency patient transport often dispatches a vehicle over several trips in a day and is concerned with its responsiveness to the rigid time windows requested by patients. However, the VRPTWMD largely considers the single trip of each vehicle and attaches more importance to the maximum routing time over the trips, because the total service time at demand sites dominates the traveling time of the servicing vehicle. Second, in the decision dimension of manpower planning, non-emergency patient transport has multi-skilled workers (ambulance operatives) with different qualifications and skills, and there can be different skill requests at different demand sites. In the VRPTWMD, the types of skills provided by the deliverymen and the skill requirements at different demand sites are the same. Meanwhile, VRPTWMD only decides the number of workers allocated to each route, without worrying about their schedule (timetable).

Manpower scheduling with skill downgrading (MSSD)

MSSD is a practical approach that involves assigning tasks to staff based on their skill levels, while allowing for the flexibility of assigning higher-skilled staff to tasks that require lower skills when necessary. Billionnet (1999), Bard (2004), Bard and Purnomo (2005), Su et al. (2023), and Dall'Olio and Kolisch (2023) all assume hierarchical skills with downgrading. Billionnet (1999) introduced an Integer Programming model that allocates agents with varying levels of skills to scheduled workdays and flexible off-days each week. This model accommodates the downgrading of agents while guaranteeing that each agent is allocated n days off within a week. Bard (2004) tackled the assignment problem for postal service workers' schedules. This approach involves allocating both full- and part-time employees to various daily shifts and rest days throughout the week. Bard and Purnomo (2005) introduced a column generation-based approach to address the preference scheduling problem for nurses, incorporating a novel aspect of skill downgrading. It focuses on optimizing the assignment of nurses with varying skill levels to shifts and days off work, considering individual preferences. Su et al. (2023) investigated a manpower allocation and vehicle routing problem with two staff qualifications and time windows, wherein staff with higher qualification levels can replace those with lower qualifications.

The study by Dall'Olio and Kolisch (2023) is most closely related to ours, as both consider multiskilled manpower with downgrading and a multi-trip VRP. Dall'Olio and Kolisch (2023) investigated a variant of the manpower scheduling and routing problem in the context of ground handling at airports, aiming to minimize the completion time of baggage loading and unloading tasks. In contrast, our objective is to develop an optimal plan for the joint optimization of vehicle routing and multiskilled manpower scheduling while minimizing the total cost. Both studies address the integration of multi-trips, time windows, and multiskilled manpower with downgrading. Nevertheless, by incorporating vehicle capacity and lunch-break constraints, our model can be regarded as an extension of theirs, thus posing additional challenges to the solution method.

Manpower scheduling and routing problem with synchronization constraints (MSRPSC)

The MSRPSC assigns, schedules, and routes a group of workers with different qualifications or skills to a set of tasks in different geographic locations to minimize the total cost of worker payment and travel expenses. Synchronization constraints are imposed on the start time of each job, which requires multiple workers, particularly those with different qualifications and skills. Team formation is a reduced form of synchronization that requires workers to gather at a central depot before traveling to customer sites (Dall'Olio & Kolisch, 2023). MSRPSC was first introduced by Li et al. (2005), who developed an arc-flow model and devised construction heuristics embedded in simulated annealing to solve this problem. Luo et al. (2016) developed a set-covering formulation for the same problem and designed an effective BPC algorithm. Cappanera et al. (2020) studied a generalization of the skill VRP in the context of home care and field service. They addressed the research problem of optimizing the routes of technicians and special devices while satisfying the requirements associated with time windows, technician skills, and the precedence and synchronization of special devices. A mixed-integer linear programming formulation was proposed and lower-bounding techniques for the proposed model were tested. Qiu et al. (2022) investigated a similar problem in the context of home healthcare. They allow patients to ask for synchronized services including at most two medical services of different skill types. If a patient requests two synchronized medical services, a predefined threshold is imposed on the difference between the starting times of the two services. Similar to the approach applied by Luo et al. (2016), they developed a set-partitioning formulation and proposed a BPC algorithm with two types of valid inequalities.

In the context of healthcare, the mathematical models and solution algorithms for MSRPSC serve the assignment and scheduling of community nurses who normally drive to patient homes to provide healthcare services during a trip. Community nurses' qualifications and skill requirements are often, if not always, the same. Community nurses are usually registered nurses (RNs) (qualification) with surgical experience and wound dressing skills. This is significantly different from the non-emergency patient transport problem investigated in this study, in which ambulances with operatives, patients, and their caregivers travel back and forth between patients' homes and hospitals. This represents a research gap between existing MSRPSC studies and our research problem.

Aircraft routing and crew pairing problem (ARCPP)

The ARCPP simultaneously determines a set of aircraft routes and crew pairings with minimum cost so that a set of flight legs is covered while satisfying a group of side constraints, such as maintenance requirements, applicable work rules of crew pairings, and minimum connection time between two consecutive flights performed by the same aircraft. Cordeau et al. (2001) originally introduced a model that jointly optimizes aircraft routing and crew pairing considering the minimum connection time constraints. A solution combining Benders' decomposition with CG was proposed. Mercier et al. (2005) extended the study of Cordeau et al. (2001) by factoring in the feasible connections between two flight legs covered by the same crew but flown in different aircrafts, which improves the robustness of the schedules. Bender decomposition was also utilized; however, the Pareto-optimal cut method was employed to speed up the solution efficiency. Weide et al. (2010) developed a monolithic formulation that integrates aircraft routing and crew pairing, improving the robustness of the solution against short flights delays. The proposed solution approach iteratively solves the crew pairing and the aircraft routing problems and increases the robustness of the solutions incrementally. Dunbar et al. (2012, 2014) minimize the expected value of propagated delay to further increase operational robustness. The ARCPP differs from the non-emergency patient transport problem under investigation in many respects. First, the transport network structure of the ARCPP does not reflect the operational characteristics of non-emergency ambulances that travel between patient homes and hospitals. Second, the ARCCP does not have to respond to diverse requests of demand sites for qualifications and skills of staff, and accordingly dispatches multiskilled staff to each transport vehicle. Third, some constraints in our study, such as capacity constraints, may not be considered in ARCPP. Therefore, it is problematic to apply mathematical formulations and solution algorithms to non-emergency patient transport problems.

VRP with manpower allocation

In recent years, the VRP combined with manpower allocation (MAVRP) in patient transport services, that is, non-emergency ambulance transfer services, has also received increasing attention. Zhang et al. (2017) determine the effective route for each non-emergency ambulance to transport patients and allocate sufficient operatives on board to satisfy the requirements of the patients. However, in their research, each ambulance only performs one trip in the planning horizon; the decision for staff planning only considers the number of staff members allocated to each route, and the specific schedule of each staff member is ignored. Lim et al. (2017) investigate a more complete research problem by incorporating many practical considerations, such as multiple trips of each ambulance, time windows of patient requests, and lunch breaks in staff scheduling. They modeled the patient transport problem as a multi-trip pickup and delivery problem with time windows and manpower planning (MTPDPTW-MP), and designed an efficient iterated local search algorithm with a variable neighborhood descent procedure. However, a patient's request for different types of facilities in an ambulance was simplified to the number of seats, and the maximum riding time of a patient on an ambulance concerning the service level was omitted. Liu et al. (2015) addressed these limitations in a so-called realistic dial-a-ride problem (R-DARP). They also contribute to Lim et al. (2017) by formulating the research problem into two mathematical models, proposing a set of valid inequalities for lower bounding, and developing a branch-and-cut algorithm as the first exact solution algorithm in this research stream. Luo et al. (2019) extended the R-DARP of Liu et al. (2015) by allowing some of the client requests to not be satisfied and changing the definition of the lunchtime windows of the staff. They developed a BPC algorithm, which was experimentally proven to be more computationally efficient than the branch-and-cut algorithm proposed by Liu et al. (2015).

Research gaps remain between existing studies on the MAVRP and the practical requirements of non-emergency patient transport services. First, these studies fail to capture patients' diverse requests for the qualifications and skills of ambulance operatives. For instance, a typical non-emergency ambulance can be manned by one or more patient transport officers (PTO). A PTO can be a driver or assistant (to paramedics) with specific qualifications. In addition to PTOs, non-emergency ambulances may also include paramedics and/or ambulance nurses, each of whom has a certain qualification and skill set. Second, these studies conducted numerical experiments mainly to evaluate the computational performance of the solution algorithms, but failed to deliver useful managerial implications/findings for guiding the operations of non-emergency patient transport services.

3. Problem description and formulations

In this section, we define the MTVRPTW-MSM problem and present two mathematical models: an arc-flow model and a trip-based set-covering model.

3.1. Problem description

Consider a non-emergency patient transport service provider that uses a group of capacitated ambulances to pick up patients from their homes within specific time windows, deliver them to the hospital, and assign a certain number of qualified staff based on patient requests. The MTVRPTW-MSM is defined on a directed graph $G = (V, A)$, where the nodes are grouped in set $V = \{0, 1, \dots, n\}$ and the arcs are grouped in set $A = \{(i, j) : i, j \in V, i \neq j \text{ or } i = j = 0\}$. $N = \{1, \dots, n\}$ represents the set of patient locations. Each patient $i \in N$ has a requirement for the number of seats a_i, service time s_i, and time window $[e_i, l_i]$, where e_i and l_i denote the earliest and latest allowable service start times for patient i, respectively. If an ambulance arrives at patient $i \in N$ before e_i, it waits until e_i starts the service. The time horizon is denoted as T. Let $[e'_0, l'_0]$ be the lunchtime window within which each staff member returns to the hospital and starts their lunch. The lunch break lasts L minutes. We assume that the staff would take lunch breaks as early as possible. If a staff member who does not have a lunch break returns to the hospital within $[e'_0, l'_0]$, they will have a lunch break of L minutes at the earliest time, instead of starting another trip before their lunch break. Normally, the lunchtime window $(l'_0 - e'_0)$ is longer than the lunch break L. The travel distance and time associated with each arc $(i, j) \in A$ are denoted by c_{ij} and t_{ij}, respectively. We assume that the triangle inequality holds for both the travel distances and travel times in terms of the arcs included in A. Let $P = \{1, 2, \dots, B\}$ be the set of skill levels and $D_m = \{1, \dots, \Gamma_m\} (m \in P)$ be the set of staff members with skill level m. Moreover, skill downgrading is allowed, that is, higher-skilled staff can serve patients who need lower-skilled staff. Assume that the skill level of p is higher than that of m and satisfies $p < m$ $(p, m \in P)$. A staff member with skill level m is paid f_m per trip. Without loss of generality, we assume that higher-skilled staff members are more expensive. Moreover, the number of staff members qualified with skill level m required by the patient $i \in N$ is denoted by d_i^m and each staff member requires one ambulance seat. A smaller value of m represents a higher level of qualification of the staff member. Let $H = \{1, \dots, K\}$ denote a set of homogeneous ambulances with capacity Q.

In the remainder of this paper, we define a trip as the journey of a vehicle traveling from the depot to serve patients and then returning to the depot, with or without a lunch break. A feasible trip comprises a sequence of patients visited, departure time from the hospital, and a decision regarding lunch, all of which must satisfy the ambulance's capacity constraints, patient time windows, and staff lunchtime window. For clarity, "trip" refers to a single journey of a vehicle, whereas "route" denotes the complete travel plan of the vehicle. We define a route as a series of feasible and sequentially ordered trips performed by the same vehicle.

3.2. Arc flow model

Based on Hernandez et al. (2014), an arc-flow model for the MTVRPTW-MSM was developed and is presented in the Appendix. The arc flow model can be directly solved using MIP solvers such as CPLEX or GUROBI. However, based on our preliminary experiments, we find that CPLEX can only solve instances of small sizes. In addition, it is widely known that the compact formulations for vehicle routing problems, such as the arc flow model presented above, tend to provide weaker lower bounds than extensive formulations, such as set-covering formulation. In the following subsection, we introduce a trip-based set-covering formulation for the MTVRPTW-MSM.

3.3. Trip-based set-covering model

Let $\mathcal{R}$ be the set of trips and $s_{mr} (m \in P)$ represent the number of staff members for type m assigned to a feasible trip $r \in \mathcal{R}$. Let α_{ir} be a binary parameter equal to 1 if patient i is served by the trip $r \in \mathcal{R}$ and 0 otherwise. Let π_{tr} be a binary parameter equal to 1 if the trip r is active at time point t and 0 otherwise. For each arc $(i, j) \in A$, β_{ijr} represents a binary parameter equal to 1 if the trip r traverse arc (i, j) and 0 otherwise. For each trip $r \in \mathcal{R}$, we define c_r as the increased cost associated with the trip r, namely, $c_r = \sum_{(i,j)\in E} c_{ij}\beta_{ijr} + \sum_{m\in P} f_m s_{mr}$. Let θ_r be a binary decision variable that equals 1 if and only if the trip r is selected. The set-covering formulation of the MTVRPTW-MSM is then given by

$$\min \sum_{r\in\mathcal{R}} c_r\theta_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} \alpha_{ir}\theta_r \geq 1, \quad \forall i \in N, \tag{2}$$

$$\sum_{r\in\mathcal{R}} \pi_{tr}\theta_r \leq K, \quad \forall t \in [0, T], \tag{3}$$

$$\sum_{r\in\mathcal{R}} s_{mr}\pi_{tr}\theta_r \leq \Gamma_m, \quad \forall m \in P, t \in [0, T], \tag{4}$$

$$\theta_r \in \{0, 1\}, \quad \forall r \in \mathcal{R}. \tag{5}$$

Objective function (1) minimizes the total cost (including travel and staff compensation costs). Constraints (2) ensure that each vertex is served exactly once. Because only K vehicles are available, constraint (3) specifies that at most K vehicles can be used at any point in time. Similarly, constraint (4) restricts the available number of staff members at each skill level. The constraint (5) defines the domain of the decision variables.

As we allow staff members with higher skill levels to serve patients requiring service from staff with lower skill levels, constraint (4) can be reformulated as follows:

$$\sum_{r\in\mathcal{R}}\sum_{b=1}^{m} s_{br}\pi_{tr}\theta_r \le \sum_{b=1}^{m}\Gamma_b, \quad \forall m\in P, t\in[0,T]. \tag{6}$$

Let S_r be the set of all possible combinations of staff numbers for trip $r\in\mathcal{R}$ with fixed-order customers. Given trip r and two combinations $S_r^1=\{s_{1r}^1,\dots,s_{mr}^1\}\in S_r$ and $S_r^2=\{s_{1r}^2,\dots,s_{mr}^2\}\in S_r$, such that $S_r^1\neq S_r^2$, we say that S_r^1 dominates S_r^2 for trip r if the following holds:

$$\sum_{b=1}^{m} s_{br}^1 \le \sum_{b=1}^{m} s_{br}^2, \quad \forall m\in P, \tag{7}$$

The inequalities in (7) signify that the combination of staff in S_r^1 does not require a higher level of skilled personnel than in S_r^2. Furthermore, the constraint (4) stipulates that for any given trip, it is essential to determine the number of staff members at each skill level $m\in P$. However, different staff combinations cannot be eliminated using the dominance rule mentioned earlier. Therefore, employing constraint (6) can significantly reduce the number of trips that must be explored, thereby enhancing the overall speed of the BPC algorithm.

Owing to the large size of set $\mathcal{R}$, it is impractical to enumerate all feasible trips and solve the set-covering model using a standard MIP solver. To solve the set-covering model precisely, in Section 5, we develop a BPC algorithm that relies on a branch-and-bound framework. At each branch and bound node, a lower bound is obtained by invoking the two-phase CG procedure introduced in Section 4, and introducing the violated valid inequalities described in Section 5.1. Moreover, because the domain of time t is continuous, the number of constraints (3) and (6) are infinite, such that the set-covering model is intractable. To overcome this difficulty, in Section 3.4, we introduce and improve a surrogate relaxation of the master problem proposed by Hernandez et al. (2016). By discretizing the time horizon into several time intervals, constraints (3) and (6) are imposed on a finite number of intervals instead of on an infinite number of time instants $[0,T]$.

3.4. Surrogate relaxation of master problem

Note that the domain of time t is continuous such that the number of constraints (3) and (6) are infinite, and the restricted linear master problem (RLMP) is intractable. To overcome this difficulty, we introduce and improve the surrogate relaxation of the master problem proposed by Hernandez et al. (2016). Let $t_{\min}$ be a predefined parameter that is less than half the traveling duration of any feasible trip. Subsequently, with $t_{\min}$ as the unit length, the time horizon $[0,T]$ is partitioned into $\lfloor \frac{T}{t_{\min}}\rfloor + 1$ time intervals grouped in $Z = \{1,\dots,\lfloor \frac{T}{t_{\min}}\rfloor+1\}$. Correspondingly, the length of the last time interval is $T - t_{\min} * \lfloor \frac{T}{t_{\min}}\rfloor$, and the others are $t_{\min}$. Thus, constraints (3) and (6) are imposed on each time interval instead of on each time instant $[0,T]$. The reformulations of constraints (3) and (6) are given by

$$\sum_{r\in\mathcal{R}} \bar{\pi}_{zr}\theta_r \le K, \quad \forall z\in Z, \tag{8}$$

$$\sum_{r\in\mathcal{R}}\sum_{b=1}^{m} s_{br}\bar{\pi}_{zr}\theta_r \le \sum_{b=1}^{m}\Gamma_b, \quad \forall m\in P, z\in Z. \tag{9}$$

where $\bar{\pi}_{zr}$, a parameter between 0 and 1, represents the proportion of time interval z covered by trip r. Therefore, the number of constraints in the primal master problem is drastically reduced (from infinite to finite).

Restricted by many more constraints, the feasible solution space of the original master problem is a subset of the feasible solution space of the surrogate relaxation of the master problem. And the feasible solution space of the surrogate relaxation of the master problem becomes closer to that of the original master problem with a finer division of the time horizon, that is, with a smaller $t_{\min}$. As illustrated in Fig. 3, S1 denotes the original problem space, while S2 and S3 represent the surrogate relaxation problem spaces, respectively. Figs. 3(a) and 3(b) respectively illustrate the scenarios where $t_{\min}^{s2}$ is and is not an integer multiple of $t_{\min}^{s3}$. If the optimal solution of the surrogate relaxation model also satisfies the original problem model, then this solution is also the optimal solution for the original problem. This principle extends to any node within the branch-and-bound tree. For a node, if a solution obtained by the surrogate relaxation model is feasible for the original master model, then it is also a solution of the original master problem. So, if the solution is the optimal solution for the node, then it is also the optimal solution for the corresponding node in the original master problem. The detailed algorithmic process to solve a node is described in Section 5.1.3.

3.5. Feasibility check

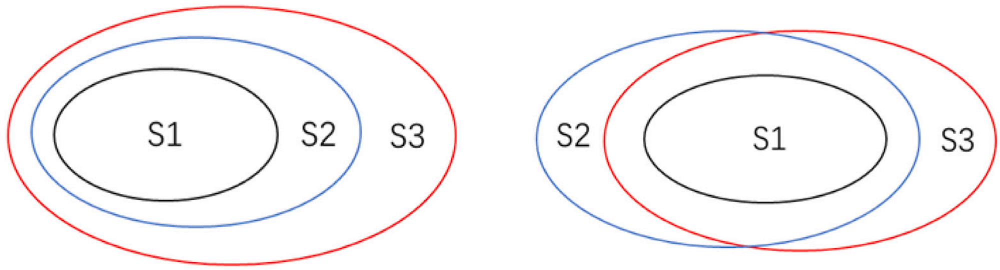
When the master problem obtains an integer solution, there may be two types of infeasible solutions, each caused by the *Surrogate Relaxation of the Master Problem* and the lack of consideration of individual staff. We explain these two scenarios separately below.

Case 1: The feasible integer solution of the surrogate relaxed master problem may be infeasible for the original master problem because the vehicle quantity constraints are not always satisfied at all time points. Next, we present a feasible solution to the surrogate master problem, which is infeasible for the original master problem.

Example 1. Consider an example involving two vehicles, as illustrated in Fig. 4. A feasible integer solution S exists that covers all customers of the surrogate relaxed master problem with four trips: r_1, r_2, r_3, and r_4. Initially, we assume that staff constraints are always satisfied, because this assumption will not affect the explanation of this example. The coverage proportions of time intervals $z=1$, $z=2$, and $z=3$ by trips r_1 and r_2 are 100%, 100%, and 33%, respectively. Similarly, for time intervals $z=3$, $z=4$, and $z=5$, the coverage proportions by trips r_3 and r_4 are 80%, 100%, 100% and 20%, 100%, 100%, respectively. We can easily see that solution S satisfies constraint (8). We observe that three trips, r_1, r_2, and r_3, overlap during the time interval $z=3$, which implies that three vehicles are required to serve these trips. However, in this example, only two vehicles are available, making solution S infeasible for the original master problem.

Case 2: Beyond the previously mentioned case, another scenario exists, as highlighted in Dall'Olio and Kolisch (2023), resulting in infeasible solutions. In the master problem, constraint (4) is replaced by constraint (6) (or (9)). These constraints focus on an aggregated level, rather than accounting for the individual circumstances of the staff. Consequently, this approach may yield unfeasible solutions. Fig. 5 illustrates an example of this phenomenon.

Example 2. Let us consider an instance with two staff members, one with level 1 and the other with level 2. There exists an integer solution S that covers all customers of the master problem with four trips: r_1, r_2, r_3, and r_4 and spans the following time intervals: $[1,2]$, $[1,4]$, $[3,6]$, and $[5,6]$, respectively. Trips r_1 and r_4 require one staff member with level 1, whereas trips r_2 and r_3 require one staff member with level 2. It is easy to see that this solution is feasible with respect to constraint (6) (or (9)). However, it is impossible to assign staff members to tours, as shown in Fig. 5. In fact, the staff member with level 1 must be assigned trips for r_1 and r_4 because of the level requirement. Consequently, this staff member cannot perform any of the other two trips because they overlap with trips r_1 and r_4. Furthermore, staff members with skill level 2 cannot be assigned to trips for both r_2 and r_3, because they overlap. Therefore, this solution to the master problem is not operationally feasible because there is no feasible assignment of the available staff to cover all trips.



(a) $t^{s2}_{\min}$ is an integer multiple of $t^{s3}_{\min}$ (b) $t^{s2}_{\min}$ is not an integer multiple of $t^{s3}_{\min}$

Fig. 3. The illustration of the original and surrogate relaxation model spaces.

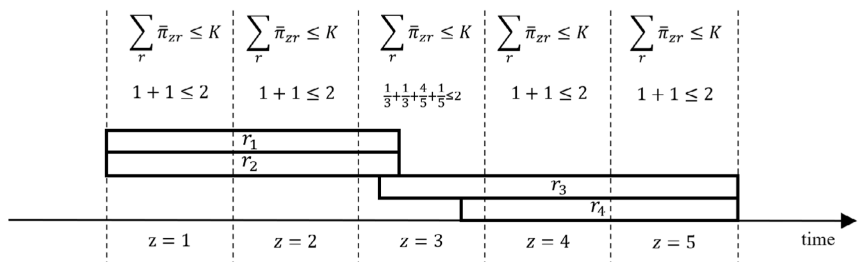


Fig. 4. Schedule of the trips of a surrogate relaxed master problem solution. Note: The inequalities show that the solution satisfies the constraints (8).

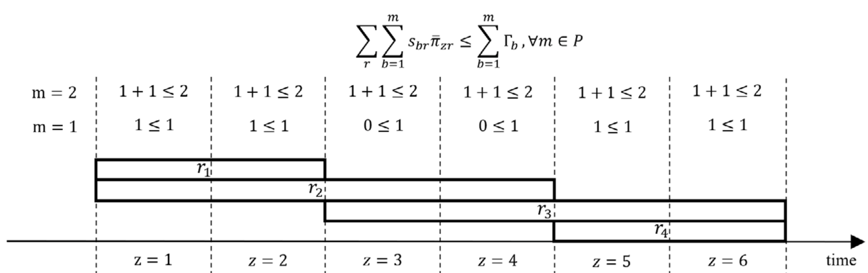


Fig. 5. Schedule of the trips of a solution with staff details. Note: The inequalities show that the solution satisfies the constraints (9).

Following the approach in Dall'Olio and Kolisch (2023), we devised a network flow problem to verify whether the master problem solution can be transformed into a plan that is operationally viable. which is called the *feasibility problem*. We introduce a graph $\bar{G} = (\bar{V}, \bar{A})$ in which each node corresponds to a trip $r \in \bar{V}$ in the solution to the master problem. Arc $(r, \tilde{r})$ in set $\bar{A}$ is only feasible if the departure time of trip $\tilde{r}$ is later than the arrival time of trip r at the depot. Let t^r_d be the departure time and t^r_e be the arrival time at the depot. Let $\bar{\mathcal{R}}$ be the set of trips required for an integer solution to the master problem. For a trip $r \in \bar{\mathcal{R}}$, the set of predecessor trips is defined as $\bar{\mathcal{R}}^r_- = \{\tilde{r} \in \bar{\mathcal{R}} | t^{\tilde{r}}_e \leq t^r_d\} \cup \{0\}$ and the set of successor tours is defined as $\bar{\mathcal{R}}^r_+ = \{\tilde{r} \in \bar{\mathcal{R}} | t^{\tilde{r}}_d \geq t^r_e\} \cup \{0'\}$, where the 0 represents the virtual starting node of every trip and $0'$ represents the virtual ending node of every trip. Our objective is to ascertain the flow of staff and trips within the graph, ensuring that there is a sufficient number of staff for each skill level to pass through each node and that each trip is sufficiently supported by sufficient vehicles at every node.

Feasibility Problem

$$\sum_{\tilde{r} \in \mathcal{R}^r_-} \sum_{b=1}^{m} \mathscr{X}^{\tilde{r},r}_b \geq \phi^r_m, \qquad \forall r \in \mathcal{R}, \forall m \in \Omega, \tag{10}$$

$$\sum_{\tilde{r} \in \mathcal{R}^r_-} \mathscr{X}^{\tilde{r},r}_m = \sum_{\tilde{r} \in \mathcal{R}^r_+} \mathscr{X}^{r,\tilde{r}}_m, \qquad \forall r \in \mathcal{R}, \forall m \in \Omega, \tag{11}$$

$$\sum_{r \in \mathcal{R}} \mathscr{X}^{0,r}_m \leq \Psi_m, \qquad \forall m \in \Omega, \tag{12}$$

$$\sum_{r \in \mathcal{R}} \mathscr{X}^{r,0'}_m \leq \Psi_m, \qquad \forall m \in \Omega. \tag{13}$$

Where Ω represents the set of resource skill levels (such as manpower or vehicles) and ϕ^r_m ($\forall m \in \Omega$) signifies the number of resources required for trip r with a minimum skill level m. In addition, $\mathscr{X}^{\tilde{r},r}_b$ denotes the number of resources with level b going from trip $\tilde{r} \in \mathcal{R}^r_-$ to trip r. Moreover, Ψ_m reflects the number of available resources at skill level m. Constraint (10) is designed to ensure the fulfillment of the resource demands for each trip. Constraint (11) is used to maintain flow conservation. Constraints (12) and (13) are applied to restrict the number of resources with skill level m both departing from and arriving at the depot, ensuring that this number does not exceed the available resources.

Under the condition that $\Omega = \{1\}$ and $\Psi_m = K$ ($m = 1$) denote the total number of vehicles, $\mathscr{X}^{\tilde{r},r}_1$ represents the number of vehicles moving from trip $\tilde{r} \in \mathcal{R}^r_-$ to trip r. In addition, $\phi^r_1 = 1$ indicates that a vehicle is essential for executing trip r. This model is useful for validating the number of vehicles in the integer solutions derived from the surrogate relaxed master problem for case 3.5. The same principle applies to the staff resources. Under the condition where $\Omega = P$, and with $\Psi_m = \sum_{b=1}^{m} \Gamma_b$ ($m \in \Omega$) signifying the total number of staff with a skill level of at least m, $\mathscr{X}^{\tilde{r},r}_m$ indicates the count of staff with at least skill level m moving from trip $\tilde{r} \in \mathcal{R}_{-}^{r}$ to trip r. Moreover, $\phi_m^r = \sum_{b=1}^{m} s_{br}$ ($m \in \Omega$) denotes the number of staff members required to execute a trip r. This model can also be employed to validate the staff count for the integer solutions obtained for the surrogate relaxed master problem in case 3.5.

Similarly, when the conditions $\Omega = P$ and $\Psi_m = \Gamma_m$ ($\forall m \in P$) are met, with $\mathscr{X}_m^{\tilde{r},r}$ representing the number of staff at level m moving from trip $\tilde{r} \in \mathcal{R}_{-}^{r}$ to trip r, and ϕ_m^r indicating the requisite number of staff for trip r with at least skill level m, this model can be used to verify the legitimacy of the number of staff in the integer solutions obtained for the master problem, as demonstrated in case 1.

4. A two-phase column generation

Considering the trip-based set-covering model as the master problem, in this section, we focus on solving the linear relaxation of the master problem (LMP) by a novel two-phase CG. In the first phase, we introduce a label-setting algorithm to obtain a set of non-dominated labels, each of which corresponds to numerous trips in terms of a considerable number of optional departure times. It is impossible to add all trips to the master problem. Therefore, in the second phase, an efficient strategy is proposed to identify the trip with the minimum reduced cost for each label.

Let RLMP be the restricted linear master problem of the set-covering formulation obtained from LMP by replacing $\mathcal{R}$ with subset $\mathcal{R}' \subseteq \mathcal{R}$. Based on the linear programming optimality condition, the optimal solution of the RLMP is also the optimal solution to the LMP if any trip $r \in \mathcal{R} \backslash \mathcal{R}'$ has a non-negative reduced cost. Therefore, the CG procedure does not directly solve the LMP using a full-trip set $\mathcal{R}$. Instead, it repeatedly solves the RLMP, passes its dual solutions to the pricing problem introduced in Section 4.1, searches for trips with negatively reduced costs by solving the pricing problem, and adds them to $\mathcal{R}'$. The CG procedure is terminated when there are no trips, and there is a negative reduced cost. This implies that the optimal value of the current RLMP, which is also the optimal value of the LMP, is obtained. For more details on CG, refer to Desaulniers et al. (2006).

4.1. Pricing problem

In the CG algorithm, the pricing problem is solved to search for trips with negatively reduced costs in terms of a given dual solution to the RLMP. In the surrogate relaxed master problem, let μ_i, λ_z ($z \in Z$) and ρ_{mz} ($m \in P, z \in Z$) be the dual variables associated with constraint (2) for patient $i \in N$, constraint (8) for the time interval $z \in Z$, and constraint (9) for the staff with at least skill level $m \in P$ and time interval $z \in Z$, respectively. The reduced cost of the variable θ_r can then be redefined as follows:

$$\bar{c}_r = c_r - \sum_{i \in N} \alpha_{ir}\mu_i - \sum_{z \in \Delta} \bar{\pi}_{zr}\lambda_z - \sum_{m \in P}\sum_{b=1}^{m} s_{br} \sum_{z \in \Delta} \bar{\pi}_{zr}\rho_{mz}. \tag{14}$$

The pricing problem can be viewed as a variant of the Elementary Shortest Path Problem with Resource Constraints (ESPPRC), which is known to be NP-complete and computationally expensive to solve optimally. Typically, the ESPPRC can be solved using a dynamic programming technique known as the label-setting algorithm. In the label-setting algorithm, labels are used to record information about a partial path starting from the source vertex 0 and ending at the patient $i \in N$. A new partial path is constructed by extending an existing feasible partial path along an arc and leaving from its end node. In principle, the algorithm constructs all the feasible partial paths starting from the initial label ℓ_0 associated with the partial path, including only the source vertex 0. To eliminate unpromising labels, a well-designed dominance rule is necessary to safely remove them without affecting the optimality of the solution. The label search is completed when the labels are extended from the source vertex to the sink vertex, constituting a complete path.

4.2. Label-setting algorithm

In this section, we provide a comprehensive introduction to the label-setting algorithm. We describe the *Feasible Time Domain* of the labels, followed by an introduction to the concept of *Optional Trips*.

4.2.1. Feasible time domain

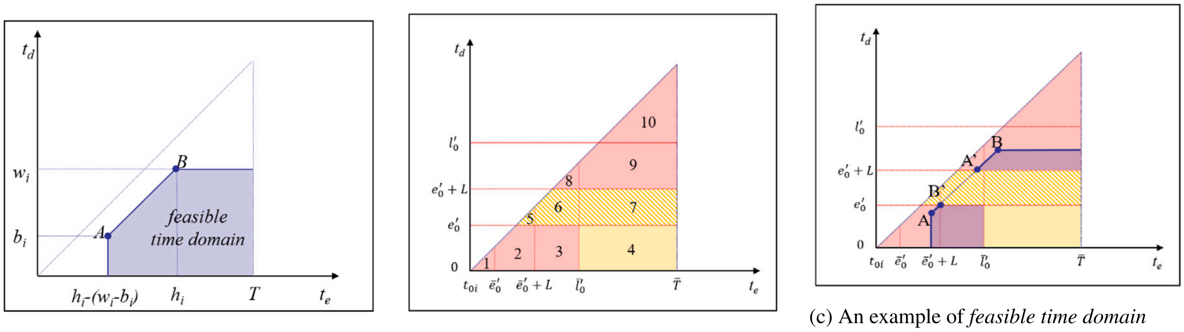
In the traditional label algorithm, it is necessary to record basic information about the partial path, such as the reduced cost, loading, departure time t_d from the depot, and service start time t_e at the end node of the partial path. For times t_d and t_e, Fig. 6(a) shows all the possible pairs of (t_d, t_e) for a specific partial path, where w_i is the latest departure time of the partial path from the hospital, h_i is the earliest service starting time at node i if the partial path leaves the hospital at time w_i, b_i is the earliest departure time of the partial path from the hospital such that its corresponding path duration is $(h_i - w_i)$. In Fig. 7, given a partial path directly from depot 0 to patient i with a time window $[e_i, l_i]$, we have $w_i = l_i - t_{0i}$, $h_i = l_i$, and $b_i = e_i - t_{0i}$. As can be seen, any point in the shaded area corresponds to one specific pair of (t_d, t_e). We refer to the shaded area in Fig. 6(a) as the *feasible time domain*, which represents the time feasible domain of the label.

Feasible time domain considering lunchtime window constraints

As mentioned in Section 3.1, each staff member should return to the hospital for a lunch break lasting L minutes. The time window for the lunch break begins at $[e_0', l_0']$. Furthermore, we assume that staff members take lunch breaks as early as possible. When a staff member not initially scheduled for a lunch break returns to the hospital within the time window $[e_0', l_0']$, it is stipulated that this individual takes a L-minute lunch break at the first available moment. This policy prioritizes staff welfare, ensuring that no staff member begins a subsequent trip without first taking their designated lunch break. Normally, the duration of the lunchtime window $(l_0' - e_0')$ is longer than that of the lunch break L. Fig. 6(b) illustrates the domain of (t_d, t_e) for all the possible partial paths ending at the patient i. Obviously, the ranges of t_d and t_e are defined by $[0, \bar{T} - t_{i0}]$ and $[t_{0i}, \bar{T}]$, respectively, where $\bar{T} = T - t_{0i} - s_i$. As shown in Fig. 6(b), the domain of (t_d, t_e) is divided into ten subdomains according to $t_d = e_0'$, $t_d = e_0' + L$, $t_d = l_0'$, $t_e = \bar{e}_0'$, $t_e = \bar{e}_0' + L$, and $t_e = \bar{l}_0'$, where $\bar{e}_0'$ and $\bar{l}_0'$ are given by $e_0' - t_{0i} - s_i$ and $l_0' - t_{0i} - s_i$, respectively. Among these ten subdomains, subdomains 4, 5, 6, and 7 (in yellow) are infeasible with respect to the lunchtime window constraints, whereas the other six (in pink) are feasible. Subdomain 4 is infeasible because the staff members allocated to the corresponding path will miss the lunchtime window. Subdomains 5, 6, and 7 contradict our assumption that the staff will take lunch breaks as early as possible, indicating that ambulances are not allowed to depart from the hospital within $[e_0', e_0' + L]$. Given the feasible subdomains shown in Fig. 6(b), we can further define the *feasible time domain* for a partial path, considering the lunchtime window constraints. By excluding the *infeasible time domain*, colored yellow, the new *feasible time domain* becomes a subset of the original *feasible time domain*. Consequently, the new *feasible time domain* comprises one or two feasible subdomains. Fig. 6(c) illustrates an example of the *feasible time domain* considering lunchtime window constraints. As shown, the *feasible time domain* is divided into two feasible subdomains by the infeasible domain of (t_d, t_e).

Correspondence between feasible time domain and feasible trips

Subsequently, we introduce the decision-making process for lunch breaks during trips, specifically, whether to allocate an L-minute lunch break to staff members upon their return to the hospital. This decision depends on the departure and arrival times at the hospital associated with each trip. Let us assume that the time horizon is segmented into four intervals based on time points e_0', $e_0' + L$, and l_0'. As depicted in Fig. 8, there are ten scenarios concerning the positions of t_d and t_e of the trips relative to the four divided intervals. These ten trip cases



(a) The time-related information in terms of a partial path.

(b) The illustration of the feasible time domain of (t_d, t_e).

(c) An example of *feasible time domain* considering the lunchtime window constraints.

Fig. 6. The illustration of *feasible time domain*.

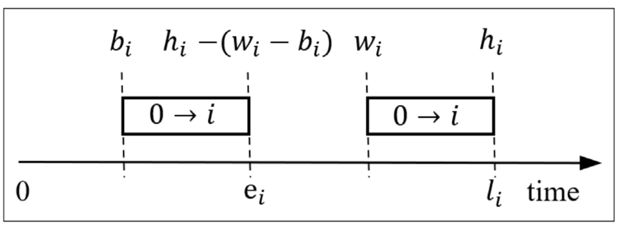


Fig. 7. An example of the symbols w_i, h_i, and b_i.

correspond to the complete paths obtained by extending the partial path in the ten subdomains represented in Fig. 6(b) to hospital 0. Specifically, scenarios 4, 5, 6, and 7 are infeasible because the staff miss the lunchtime window in scenario 4, whereas scenarios 5, 6, and 7 contradict our assumption that the staff will take lunch breaks at the earliest opportunity. Among the remaining six scenarios, the staff members in scenarios 2 and 3 are scheduled for lunch breaks. Staff members returning to the hospital before e'_0 in scenario 1, if they are assigned to a subsequent trip, such as scenarios 2 or 3, can have a lunch break upon their return; if assigned to a trip, as in scenarios 8, 9, or 10, they will have sufficient time for a lunch break. Similarly, staff members departing from the hospital after $e'_0 + L$ in scenarios 8, 9, or 10, if they are assigned to a previous trip, such as scenarios 2 or 3, they can have a lunch break upon their return; if assigned to a trip, such as scenario 1, they will have sufficient time for a lunch break. Thus, each staff member can receive a lunch break within the lunchtime window constraints.

4.2.2. Optional trips

This subsection introduces the concept of *optional trips*, which are trips with the potential to achieve a minimal reduction in cost. To better illustrate this concept, we first explore the notion of *minimal traveling duration trips* in the MTVRPTW, a subset of *optional trips* characterized as *minimal traveling duration trips*.

Minimal Traveling Duration Trips, The reduced cost of a trip, as defined in Eq. (14), is not only dependent on the set of visited patients and their traveling cost, but also on the departure and arrival times at the depot. As dual variables, λ_z and ρ_{mz} are nonpositive. Given a set of sequential patients, a trip with a shorter traveling duration may have a lower value of the reduced cost than one with a longer traveling duration. We allow the ambulance to wait for patient $i \in N$ if it arrives at the patient's location before the earliest allowable service start time for that patient, e_i. Therefore, to reduce the duration as much as possible, we should avoid unnecessary waiting times for patients by advancing the departure time from the depot, provided that the trip is not rendered infeasible. Thus, we can reduce the traveling duration of a trip, and in turn, decrease its reduced cost. Hence, in the traditional MTVRPTW, we can always find a *minimal traveling duration trip* that achieves a minimal reduced cost for a given set of sequential patients. As shown in Fig. 9, trips whose (t_d, t_e) lie on the line segment AB have *minimal traveling duration*. That is, the *minimal traveling duration trips* must be on line segment AB. In fact, in the traditional MTVRPTW, there exists an an optimal solution, where all trips in this solution are *minimal traveling duration trips*. For the traditional MTVRPTW, it is sufficient to search for trips with *minimal traveling duration* during the column generation process.

However, in our problem, the aforementioned property of the traditional MTVRPTW may no longer hold because of the inclusion of lunchtime window constraints. As introduced in Section 4.2.1, the *feasible time domain* of a path considering lunchtime window constraints is a subset of the original *feasible time domain*. The *feasible time domain* may consist of one or two feasible subdomains. The *feasible time domain*, which includes two feasible subdomains, must be addressed separately. For each feasible subdomain, we can derive a group of *optional trips* that are promising for achieving minimal cost reduction. We discuss this by considering two classical cases as examples, as shown in Figs. 10 and 11. In the first case, the line segment AB is split into two sub-segments, AB' and $A'B$, as seen in Fig. 10. In this scenario, the (t_d, t_e) of the *optional trips* must be either on the sub-segment AB' or on $A'B$, indicating the *minimal traveling duration trips*. However, in the second case, the trips may achieve a minimal reduced cost when their (t_d, t_e) does not lie on line segment AB (*minimal traveling duration*). As shown in Fig. 11, the *feasible time domain* is split into two feasible subdomains. Interestingly, one of these subdomains does not intersect line segment AB (purple rectangle at the bottom of the left subplot in Fig. 11). Within the specified subdomain, the unique trip located at point B' in the upper left corner, while not being the trip with the minimal traveling duration, has the potential to achieve the minimal reduced cost. Therefore, the (t_d, t_e) of *optional trips* will lie either on line segment $A'B$ or at point B'. It is important to note that the traveling duration of the trip corresponding to point B' is minimal among the trips in this subdomain, but longer than the minimal duration of trips whose (t_d, t_e) lies on the line segment AB. Points B' and A correspond to the same arrival times at the hospital; however, point B' indicates an earlier departure time than point A. In the case of the *feasible time domain*, including one feasible subdomain, the solution is a special case of a situation with two *feasible time domains*.

In the subsequent sections of this article, all trips mentioned are referred to as *optional trips*. There definitely exists an optimal solution, where all trips in this solution are *optional trips*. Therefore, during the column generation process, we only need to search for the *optional trips* for MTVRPTW-MSM. Next, we will introduce the label algorithm process, where each label corresponds to a set of *optional trips*, as shown on the right in Figs. 10 and 11.

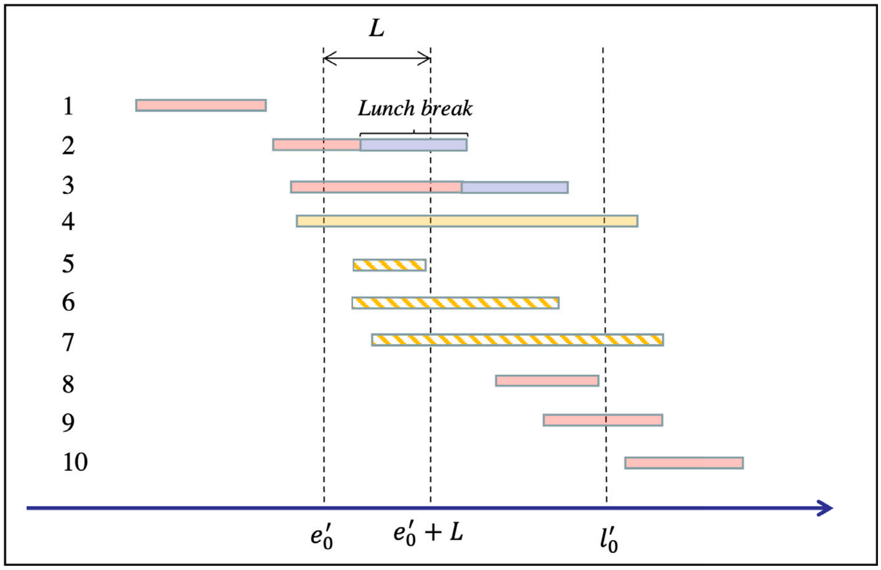


Fig. 8. The illustration of the lunch break decision.

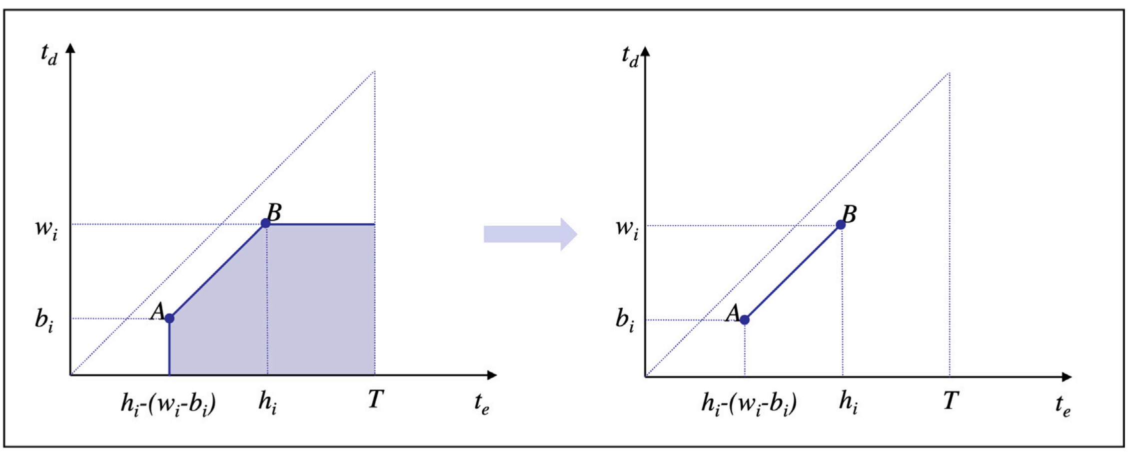


Fig. 9. The trips with the minimal traveling duration have the potential to achieve minimal reduced cost for traditional MTVRPTW.

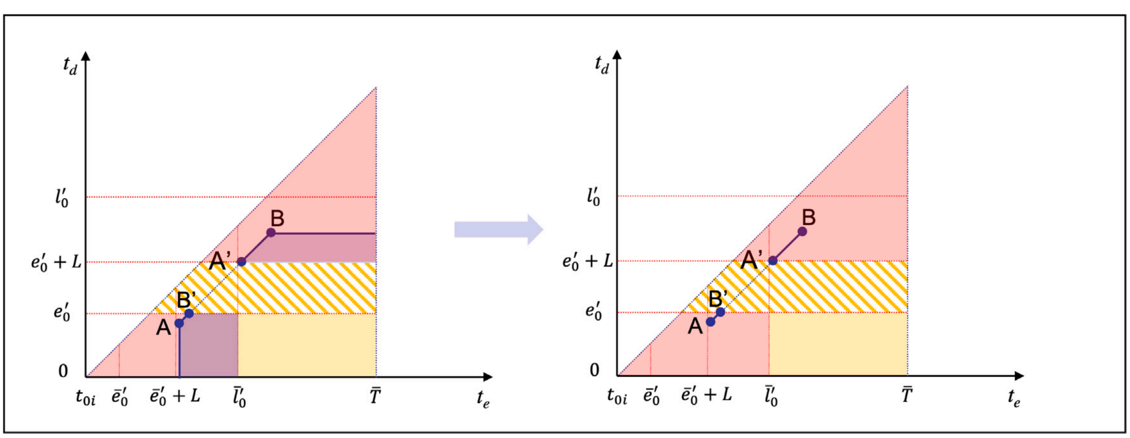


Fig. 10. The trips have the potential to achieve minimal reduced cost considering lunchtime window constraints. (a).

4.2.3. Label definition

Specifically, we denote a partial path Ξ_i from hospital to patient i by a label $\ell_i = \{fc_i, P_i = \{p_i^1, \ldots, p_i^B\}, q_i, b_i, w_i, h_i, V_i, N_i\}$. The meaning of each component of the label is presented as follows:

- fc_i: the reduced cost derived from the summation of the travel cost and the dual variables of visited patients;
- P_i: the set of the number of staff members with different levels equipped in the path, where p_i^b is the total number of the first b kinds of staff members;

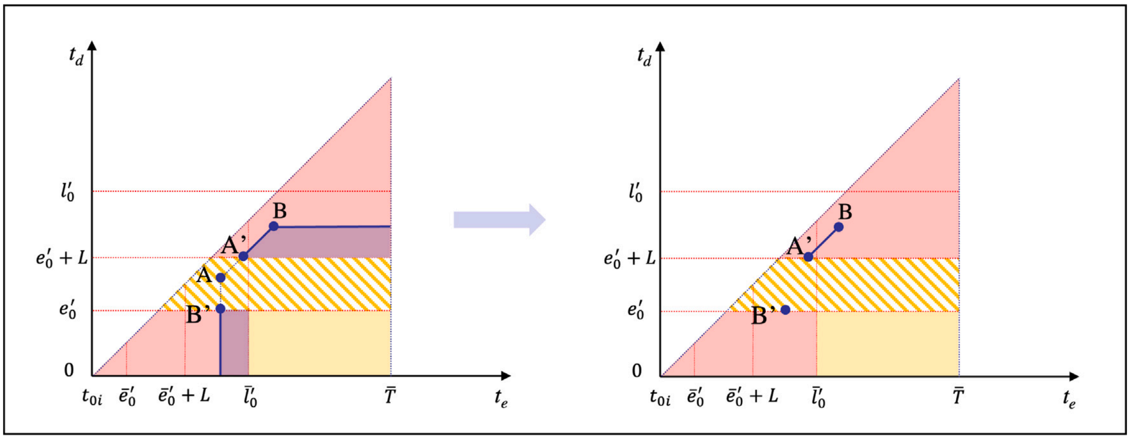


Fig. 11. The trips have the potential to achieve minimal reduced cost considering lunchtime window constraints. (b).

- q_i: the number of seats occupied along path Ξ_i;
- w_i: the latest departure time of path Ξ_i from the hospital;
- h_i: the earliest service starting time at node i if path Ξ_i leaves the hospital at time w_i;
- b_i: the earliest departure time of path Ξ_i from the hospital such that its corresponding trip duration is $(h_i - w_i)$;
- V_i: the set of patients that can be expanded from path Ξ_i. Note that path Ξ_i can be extended to a patient only if the patient has not been visited by the path as well as if the vehicle capacity and time window constraints are satisfied;
- N_i: the set of patients that have been visited by path Ξ_i.

4.2.4. Label extension functions and dominance rule

The initial label is defined by $\ell_i = (0, \emptyset, 0, e_0, l_0, l_0, N, \emptyset)$. Given a label $\ell_i = (fc_i, P_i, q_i, b_i, w_i, h_i, V_i, N_i)$, we can obtain another label ℓ_j by extending ℓ_i to $j \in V_i$ through arc (i, j) according to the following extension functions:

$$fc_j = fc_i + c_{ij} - \mu_j, \tag{15a}$$

$$p_j^m = \max\{p_i^m, \sum_{b=1}^{m} d_j^b\}, \quad \forall m \in P, \tag{15b}$$

$$q_j = q_i + a_j + p_j^B - p_i^B, \tag{15c}$$

$$w_j = \min\{w_i, w_i - (h_i + s_i + t_{ij} - l_j)\}, \tag{15d}$$

$$h_j = \max\{e_j, \min\{l_j, h_i + s_i + t_{ij}\}\}, \tag{15e}$$

$$b_j = \min\{w_j, \max\{b_i, e_j - (h_i + s_i + t_{ij} - w_i)\}\}, \tag{15f}$$

$$N_j = N_i \cup \{j\}, \tag{15g}$$

$$V_j = V_i \backslash \{j\} \backslash \{k \in V_i : a_k + q_j + \max\{p_j^B, \sum_{m=1}^{B} d_k^m\} - p_j^B > Q\}$$
$$\backslash \{k \in V_i : h_j - (w_j - b_j) + s_j + t_{jk} > l_k\}. \tag{15h}$$

Proposition 1 (*Dominance Rule*). *Suppose that two labels $\ell_i^1 = (fc_i^1, P_i^1, q_i^1, b_i^1, w_i^1, h_i^1, V_i^1, N_i^1)$ and label $\ell_i^2 = (fc_i^2, P_i^2, q_i^2, b_i^2, w_i^2, h_i^2, V_i^2, N_i^2)$ end at the same node i, ℓ_i^1 dominates ℓ_i^2 if the following rules hold:*

1. $fc_i^1 \le fc_i^2$,
2. $p_i^{m1} \le p_i^{m2}, \quad \forall m \in P$,
3. $q_i^1 - p_i^{B1} \le q_i^2 - p_i^{B2}$,
4. $h_i^1 - w_i^1 \le h_i^2 - w_i^2$,
5. $h_i^1 + b_i^1 - w_i^1 \le h_i^2 + b_i^2 - w_i^2$,
6. $w_i^1 \ge w_i^2$,
7. $V_i^1 \supseteq V_i^2$.

Due to space limitations, the proof of Proposition 1 has been included in the appendix.

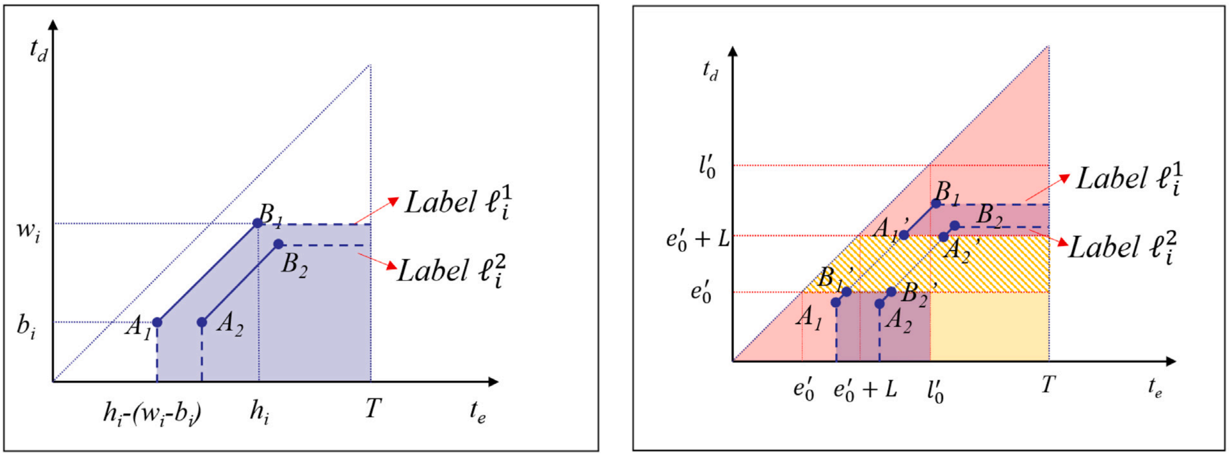
The time-related dominance rule is illustrated in Fig. 12. As can be observed from Fig. 12(a), if label ℓ_i^1 dominates ℓ_i^2, then the *feasible time domain* of ℓ_i^2 is a subset of that of ℓ_i^1. However, we did not mention the lunchtime window constraints when we discussed the label extension function and dominance rules. It can be easily observed that the label extension function and dominance rule will still hold when we take the lunchtime window constraints into consideration. As shown in Fig. 12(b), excluding the *infeasible time domain* caused by the lunchtime window constraints, the *feasible time domain* of ℓ_i^2 is still the subset of the *feasible time domain* of ℓ_i^1.

4.3. Identify an optional trip with minimal reduced cost

In this section, we present the second part of the two-phase CG. According to the label-setting algorithm in Section 4.2, we can obtain a set of non-dominated labels in terms of complete paths. Each label records a set of ordered patients, corresponding to a set of *optional trips*. Therefore, there could be a large number of feasible combinations for (t_d, t_e), which might lead to the creation of a significant number of *optional trips* with negative reduced costs. In each iteration of the CG procedure, as for a specific label, we can add multiple *optional trips* with a negative reduced cost or one with a minimal reduced cost. In this work, we employ the latter one since our preliminary experiments show that the latter method is more effective than the former one. Below, we develop a procedure to identify an *optional trip* with the minimal reduced cost associated with a label.

According to the above procedure, in terms of each complete path, we can derive one or two groups of *optional trips* likely to achieve minimal reduced cost. Then, we will further identify a specific trip with minimal reduced cost among these *optional trips*. If we have two groups of optional trips, we need to first find two trips (one for each group) with minimal reduced cost and then select the one with a lower reduced cost. Next, we will discuss the process of searching for a trip with the minimum reduced cost from one group of *optional trips*. These trips correspond to a segment or point that was mentioned in Section 4.2.2. To provide further clarification, a segment or point can be represented by a tuple (t_b, t_w, t_h), where t_b denotes the earliest departure time among the group of *optional trips*, t_w denotes the latest departure time among the group of optional trips, and t_h denotes the earliest arrival time if the vehicle returns to the hospital at time t_w. It is important to keep in mind that this tuple applies to both segments and points, with points being a special case of segments in this context.

Based on the calculation of reduced cost (14), the term $c_r - \sum_{i \in N} \alpha_{ir}\mu_i$ is constant for a given set of ordered patients. The other two terms in the reduced cost are referred to as time-dependent reduced cost tc_r, which can be reformulated as $tc_r = -\sum_{z \in Z} \bar{\pi}_{zr}\lambda_z - \sum_{m \in P}\sum_{b=1}^{m} s_{br} \sum_{z \in Z} \bar{\pi}_{zr}\rho_{mz} = -\sum_{z \in Z} \bar{\pi}_{zr}(\lambda_z + \sum_{m \in P}\sum_{b=1}^{m} s_{br}\rho_{mz}) = -\sum_{z \in Z} \bar{\pi}_{zr}\sigma_z$.



(a) The illustration of time-related dominance rule for traditional MTVRPTW

(b) The illustration of time-related dominance rule considering lunchtime window constraints

Fig. 12. The illustration of time-related dominance rule.

Meanwhile, σ_z can be considered as a given parameter only related to the information about staff requirement of the visited patients s_{br} as well as the dual solution of λ_z and ρ_{mz}. It shows that the reduced cost of an *optional trip* in terms of the complete path relies on the time intervals covered by it. As mentioned in Section 3.4, the unit length of time interval $z \in Z$ is denoted by $t_{\min}$. In addition, let $d_{\min}$ denote the *minimal traveling duration*. So, the reduced cost of the optional trips only depends on the departure time at the depot. We denote $tc_r(t_d)$ as a function representing the time-dependent reduced cost of a path with respect to t_d, the departure time at the depot. The function $tc_r(t_d)$ can be defined as follows:

$$tc_r(t_d) = \int_{t_d}^{t_d+d_{\min}} \sigma'(t)dt \tag{16}$$

where $\sigma'(t) = -\sigma_z \; \forall t \in [z_{start}, z_{end}), z \in Z$ is a non-monotonic step function, where the jump discontinuities are the start instant z_{start} or the end instant z_{end} of time intervals $z \in Z$. Then, $tc_r(t_d)$ is a non-monotonic piecewise linear function. Thinking of t_d ranging from the earliest departure time t_b to the latest departure time t_w, the minimum reduced cost will be found as the minimum of the following four cases:

- t_d (the lower limit of the integral in (16)) equals to one of jump discontinuities of $\sigma'(t)$;
- $t_d + d_{\min}$ (the upper limit of the integral in (16)) equals to one of jump discontinuities of $\sigma'(t)$;
- t_d equals to t_b;
- t_d equals to t_w.

We can identify an *optional trip* with minimal reduced cost by comparing the reduced costs associated with the departure times in these four cases. Fig. 13 illustrates an example for searching for the trip with the minimum reduced cost given a specific complete path Ξ. Figs. 13(a)–13(c) illustrate the four cases introduced above, the function $\sigma'(t)$, and the function $tc_r(t_d)$, respectively. Suppose the length of each time interval $t_{\min}$ and the minimal traveling duration $d_{\min}$ of path Ξ are 2 and 4.4, respectively. The earliest departure time t_b and the latest departure time t_w at the depot are assumed as 2.6 and 4.6, respectively. In addition, we assume the parameter σ_z ($z \in \{1,2,3,4,5\}$) to be $\sigma_1 = -1$, $\sigma_2 = -3$, $\sigma_3 = 0$, $\sigma_4 = -1$, and $\sigma_1 = -2$. The departure time and its related time-dependent reduced cost are shown in Fig. 13(c) as numbers in brackets. As can be seen, the trip with minimum reduced cost is the one whose departure time t_d equals $2t_{\min}$.

4.4. Accelerating strategies

We develop three techniques to speed up further the pricing problem solution method, namely bidirectional search, heuristic pricing, and *ng*-route relaxation. In the following, we will introduce the accelerating strategies in detail.

4.4.1. Bidirectional search

In the bidirectional search strategy, the labels are extended forward from the source vertex 0 and backward from the sink vertex 0, and then pairs of forward and backward labels ending at the same vertex are connected together to be a complete trip. We choose time as the key resource to stop the path extension in both forward and backward directions. Specifically, we only extend a partial path to a node whose starting time is less than or equal to half of the time horizon, *i.e.*, $T/2$.

We denote a backward partial path Ξ_i^b from node i to hospital 0 by a label $\ell_i^b = (fc_i^b, P_i^b = \{p_i^{1b}, \ldots, p_i^{Bb}\}, q_i^b, b_i^b, w_i^b, h_i^b, V_i^b, N_i^b)$, where parameters fc_i^b, P_i^b, q_i^b, V_i^b, and N_i^b have the same meanings as their counterparts in the forward label ℓ_i. The meanings of other parameters are as follows:

- b_i^b: the latest time that path Ξ_i^b arrives at the hospital;
- w_i^b: the earliest time that path Ξ_i^b arrives at the hospital;
- h_i^b: the service starting time at node i if the arrival time at the hospital is w_i^b.

A backward label is initially set as $\ell_0^b = \{0, \emptyset, 0, l_0, 0, 0, N, \emptyset\}$. Label ℓ_i^b can be extended to node $j \in V_i^b$ through arc (j,i), producing a new label ℓ_j^b based on the following extension functions:

$$fc_j^b = fc_i^b + c_{ji} - \mu_j, \tag{17a}$$

$$p_j^{mb} = \max\{p_i^{mb}, \sum_{b=1}^{m} d_j^b\}, \; \forall m \in P, \tag{17b}$$

$$q_j^b = q_i^b + a_j + p_j^{Bb} - p_i^{Bb}, \tag{17c}$$

$$b_j^b = \min\{b_i^b, b_i^b + \min\{0, l_j - h_i^b + s_j + t_{ji}\}\}, \tag{17d}$$

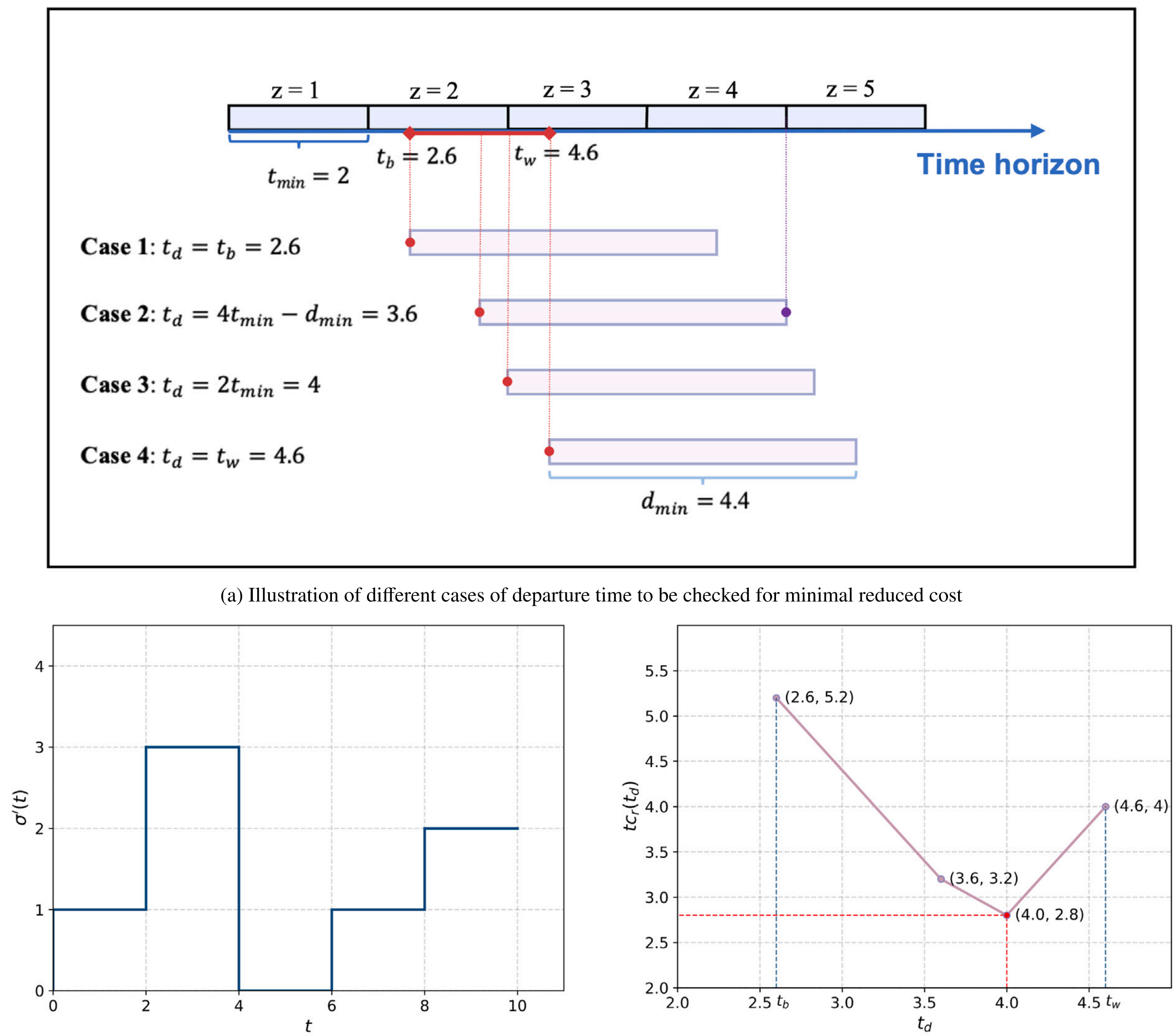
$$w_j^b = \max\{w_i^b, e_j + t_{ji} + s_j + w_i^b - h_i^b\}, \tag{17e}$$

$$h_j^b = \min\{l_i, \max\{e_j - h_i^b + s_j + t_{ji}, 0\} + h_i^b - s_j - t_{ji}\}, \tag{17f}$$

$$N_j^b = N_i^b \cup \{j\}, \tag{17g}$$

$$V_j^b = V_i^b \backslash \{j\} \backslash \{k \in V_i^b : a_k + q_j^b + \max\{p_j^{Bb}, \sum_{m \in P} d_k^m\} - P_j^{Bb} > Q\}$$
$$\backslash \{k \in V_i^b : b_j^b - s_k - t_{kj} < e_k\}. \tag{17h}$$

Proposition 2 (*Backward Dominance Rule*). *Given two labels ℓ_i^{b1} and ℓ_i^{b2} ending at the same node i, ℓ_i^{b1} dominates ℓ_i^{b2} if the following conditions are satisfied:*



(a) Illustration of different cases of departure time to be checked for minimal reduced cost

(b) Function $\sigma'(t)$ w.r.t. time t covered by path Ξ

(c) Function $tc_r(t_d)$ w.r.t. departure time t_d of path Ξ

Fig. 13. Searching for an *optional trip* with minimum reduced cost.

1. $fc_i^{b1} \le fc_i^{b2}$,
2. $p_i^{mb1} \le p_i^{mb2}$, $\forall m \in P$,
3. $q_i^{b1} - p_i^{Bb1} \le q_i^{b2} - p_i^{Bb2}$,
4. $w_i^{b1} - h_i^{b1} \le w_i^{b2} - h_i^{b2}$,
5. $h_i^{b1} + b_i^{b1} - w_i^{b1} \ge h_i^{b2} + b_i^{b2} - w_i^{b2}$,
6. $w_i^{b1} \le w_i^{b2}$,
7. $V_i^1 \supseteq V_i^2$.

Given a forward label associated with node i and a backward label associated with node j, we can combine ℓ_i^f and ℓ_j^b if the following conditions are satisfied:

- $V_i^f \supseteq N_j^b$,
- $V_j^b \supseteq N_i^f$,
- $q_i^f + q_j^b - \min\{p_i^{Bf}, p_j^{Bb}\} \le Q$,
- $b_i^f + h_i^f - w_i^f + s_i + t_{ij} \le b_j^b + h_j^b - w_j^b$.

Correspondingly, the resulting reduced cost associated with traveling cost and the dual variables of visited customers is obtained by $fc = fc_i^f + fc_j^b + c_{i,j}$.

4.4.2. Heuristic pricing

We develop a heuristic labeling algorithm for the resolution of the pricing problem. Specifically, each time when apply the forward and backward dominance rules, we first ignore the sub-set row inequalities and the conditions related to the set of nodes that a partial path can be extended to, *i.e.*, set V_i and V_i^b related conditions. When no feasible trip is found, we run another heuristic labeling algorithm that simply ignores the set of node conditions.

4.4.3. Ng -route relaxation

The pricing problem is a variant of ESPPRC, which requires each vertex to be visited at most once. ESPPRC is a well-known NP-hard problem that is hardly solved effectively. As such, various studies have developed an accelerated strategy based on the relaxation of the ESPPRC. In most situations, the elementary requirements are typically completely or partially relaxed. Currently, state-of-the-art studies often apply the *ng*-route relaxation attempting to achieve a better balance between pricing routes effectively and finding good lower bounds. The *ng*-route relaxation is originally introduced by Baldacci et al. (2011) to solve the capacity vehicle routing problem (CVRP). In this study, we also utilize this method to speed up our label-setting algorithm.

For each patient $i \in N$, a neighborhood NB_i (referred to as *ng*-set) is defined by its nearest ϵ patients, including patient i itself. The *ng*-route relaxation method aims to construct the *ng*-routes that do not necessarily satisfy the elementary constraint. An *ng*-route is defined by a trip without containing any *ng*-cycle $(v_i, \dots, v_j)$, which satisfies both the conditions $v_i = v_j$ and $v_j \in \cap_{k=i,\dots,j-1} NB_{v_k}$. It means that the visit to a vertex i might be "neglected" and the path can be extended to i again, producing a cycle. However, the cycle is only allowed over the vertex i when there exists a node u in this cycle such that $i \notin NB_u$.

Given a path $\Xi = (v_1, v_2, \dots, v_n)$, let $V(\Xi)$ and $\Pi(\Xi)$ be the set of patients visited in path Ξ and the set of patients to which are forbidden to be extended. Then the set $\Pi(\Xi)$ can be defined as follows:

$$\Pi(\Xi) = \{v_k \in V(\Xi)\backslash\{v_m\} : v_k \in \cap_{s=k+1}^{m} NB_{v_s}\} \cup \{v_m\} \tag{18}$$

we can extend the label $\ell(\Xi)$ to vertex v_{n+1} only if $v_{n+1} \notin \Pi(\Xi)$ as well as the other resource constraints are satisfied. Let *ng*-SSPRC denote the *ng*-route relaxation of ESSPRC. The *ng*-SSPRC can be considered as a compromise between ESSPRC and SSRPC. Note that the parameter ϵ, also known as the cardinality of *ng*-set, is critical to determine the trade-off between overall computational efficiency and solution quality. The larger the ϵ, the *ng*-SSPRC is closer to ESSPRC. As ϵ increases from 0 to $|N| - 1$, the *ng*-SSPRC will become from SSPRC to the ESSPRC. Therefore, to achieve a reasonable balance between computing efficiency and solution quality, exploratory experiments are required to discover a suitable value of ϵ. Based on some preliminary tests, we set ϵ to be 5. To further accelerate *ng*-route relaxation method, we combine the Decremental State-Space Relaxation (DSSR) technique with *ng*-route relaxation, which is originally proposed by Martinelli et al. (2014). In the implementation of the bidirectional search label-setting algorithm, the label components V_j and V_j^b are updated as follows:

$$V_j = V_i \cup (N\backslash NB_j)\backslash\{j\}\backslash\mathbb{N}_q\backslash\mathbb{N}_t \tag{19}$$

$$V_j^b = V_i^b \cup (N\backslash NB_j)\backslash\{j\}\backslash\mathbb{N}_q^b\backslash\mathbb{N}_t^b \tag{20}$$

where $\mathbb{N}_q$ and $\mathbb{N}_t$ refer to subsets of customers to which the forward label ℓ_j cannot extend, since they can potentially cause violations of vehicle capacity constraints and customer time window constraints, respectively. Subsets $\mathbb{N}_q^b$ and $\mathbb{N}_t^b$ have similar definitions to $\mathbb{N}_q$ and $\mathbb{N}_t$ but are defined for backward labels. These four subsets are computed as:

$$\mathbb{N}_q = \{k \in N : a_k + q_j + \max\{p_j^B, \sum_{m=1}^{B} d_k^m\} - p_j^B > Q\} \tag{21}$$

$$\mathbb{N}_t = \{k \in N : h_j - (w_j - b_j) + s_j + t_{jk} > l_k\} \tag{22}$$

$$\mathbb{N}_q^b = \{k \in N : a_k + q_j^b + \max\{p_j^{Bb}, \sum_{m\in P} d_k^m\} - P_j^{Bb} > Q\} \tag{23}$$

$$\mathbb{N}_t^b = \{k \in N : b_j^b - s_k - t_{kj} < e_k\} \tag{24}$$

5. Branch-and-price-and-cut algorithm

To obtain the optimal integer solution, we develop a BPC algorithm, which is well-known to be used to solve mixed integer programming with a large number of variables. Over the B&B procedure, the CG algorithm introduced in Section 4 is used to solve the LMP of each B&B node, whose optimal value can be viewed as a valid lower bound. If the B&B node is not fathomed, the valid inequalities are separated and added to the LMP, which is known as the cutting plane approach. Then, the CG procedure is invoked again. In this way, the separation of valid inequalities and CG procedure is run alternatively until no valid inequalities can be found. Note that the identified valid inequalities will be retained in the rest of the B&B nodes. In the rest of this section, we will introduce the valid inequalities designed for the MTVRPTW-MSM and the branching strategy utilized to explore the B&B tree.

5.1. Valid inequalities

In this subsection, we present two groups of valid inequalities, namely staff-based inequalities and subset row inequalities, which are used to improve the lower bound of the LMP at each B&B node. For ease of simplicity, we only introduce the combination of these inequalities with a forward label-setting algorithm in detail, and it can be easily extended to a bidirectional search with small modifications.

5.1.1. Staff-based inequalities

The staff-based inequalities are derived from the well-known k-path inequalities, which can efficiently improve the lower bound of the linear relaxation of vehicle routing problems with time windows. Each of them is used to limit the minimum number of vehicles that can serve a particular group of customers. Let $E^-(S) = \{(i,j) | i \notin S, j \in S\}$ denote the set of arcs entering the specified subset S. The staff-based inequalities can be described as follows:

$$\sum_{r\in R} \sum_{(i,j)\in E^-(S)} \beta_{ijr}\theta_r \ge \Theta_S, \ \forall S \subset N, |S| > 2, \tag{25}$$

where the β_{ijr} is a binary parameter, which equals 1 if trip r passes through arc (i,j). Θ_S represents the lower bound of the number of trips entering the specified subset S. It can be calculated according to the algorithm 1. Let $\{\eta_S\}, S \subset N$ be the dual variables of constraints (25). To combine the staff-based inequalities into the label-setting algorithm, we modify the extension function of fc_j as $fc_j = fc_i + c_{ij} - \mu_j \sum_{S\in N} \sum_{(i,j)\in E^-(S)} \beta_{ijr}\eta_S$.

To separate the staff-based inequalities efficiently, we employ two separation approaches, namely the trip-based separation method (Archetti et al., 2011) and the partial enumeration heuristic (Desaulniers, 2010). Specifically, the trip-based separation method is employed first, and if it fails to find any violated inequality, we then invoke the partial enumeration heuristic to further search for the violated staff-based inequalities. Since the partial enumeration heuristic is computationally expensive, to strike a balance between computational efficiency and solution quality, we only invoke it at the B&B nodes whose depth is less than 6. On the contrary, the trip-based separation method is utilized at each B&B node explored in the B&B procedure.

Algorithm 1: Calculating the lower bound of the number of trips entering the specified subset S

1: $\vartheta \leftarrow \left\lceil \frac{\sum_{i\in S} a_i}{Q} \right\rceil$, $\Phi_i \leftarrow$ The number of staff required to serve customer $i \in S$, $\Phi = \{\Phi_1, \dots, \Phi_{|S|}\}$ (sort in ascending order)
2: $\varrho \leftarrow \sum_{i=1}^{\vartheta} \Phi_i$, $\bar{\vartheta} \leftarrow \left\lceil \frac{\sum_{i\in S} a_i + \varrho}{Q} \right\rceil$
3: **while** $\vartheta \ne \bar{\vartheta}$ **do**
4: $\quad \varrho \leftarrow \varrho + \sum_{i=\vartheta+1}^{\bar{\vartheta}} \Phi_i$, $\vartheta \leftarrow \bar{\vartheta}$, $\bar{\vartheta} \leftarrow \left\lceil \frac{\sum_{i\in S} a_i + \varrho}{Q} \right\rceil$
5: **end while**
6: **return** ϑ

5.1.2. Subset row inequalities

To further improve the quality of lower bounds, we adopt the subset row inequality originally introduced by Jepsen et al. (2008). Specifically, for a subset $S \subset N$ and an integer s $(1 < s \le |S|)$, the subset row inequalities are expressed as follows:

$$\sum_{r\in R} \left\lfloor \frac{1}{s} \sum_{i\in S} \alpha_{ir} \right\rfloor \theta_r \le \left\lfloor \frac{|S|}{s} \right\rfloor. \tag{26}$$

By varying the size of S and the parameter s, different families of subset row inequalities can be derived. In line with the literature, we only consider $|S| = 3$ and $s = 2$. To separate violated inequalities, we utilize a straightforward method to enumerate all of the subset row inequalities. Since the subset row Inequalities make the subproblem more difficult to solve, we only enumerate them at the B&B nodes whose depth is not greater than 2.

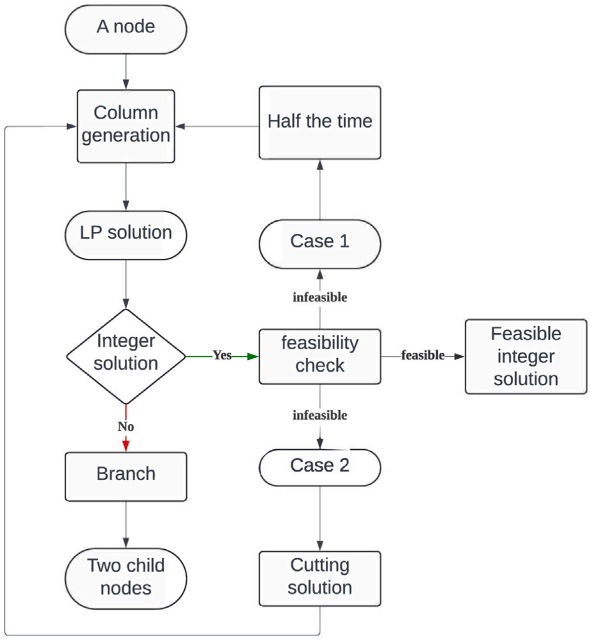


Fig. 14. The detailed algorithmic process to solve a node.

5.1.3. Cutting infeasible solutions

While exploring the branch-and-bound tree, when an optimal integer solution is reached at a particular node, we need to consider two scenarios to determine if this solution is feasible to cut infeasible solutions, as explained in Section 3.5. To verify this integer solution, we assess its validity in two cases by solving the *Feasibility Problem*. If this solution is feasible, then the integer solution at this node is feasible for the original problem, and we can terminate our exploration of this node because we have found the optimal solution for this node. Otherwise, if this integer solution is not a valid solution to the original problem, as in feasibility check case 3.5—typically due to overlapping vehicles or staff with at least level m—we must reduce the time interval length, denoted as $t_{\min}$, by halving it. Subsequently, we re-solve the node using the column generation algorithm. In feasibility check case 1, we can add a solution cut to the master problem to prohibit the current integer solution and then re-solve the node using the column generation algorithm. The detailed algorithmic process to solve a node is shown in Fig. 14. In the BPC algorithm, we only perform branching on arcs, so every node on the branch-and-bound tree of the relaxed problem corresponds one-to-one with the nodes of the original problem. Reducing the time $t_{\min}$ by half or cutting off infeasible solutions does not affect the optimal solution of the original problem's node. When an integer optimal solution of a node in the relaxed problem is found to be feasible for the original problem through algorithm shown in Fig. 14, we can safely assume that the optimal solution for the corresponding node in the original problem has been found. When the LP solution is non-integral, we branch to create two new nodes, which correspond to two nodes in the original problem, and algorithm shown in Fig. 14 can be repeated to solve them. The algorithm runs iteratively in this way until an optimal solution is obtained.

Let $\bar{\mathcal{R}}$ be the set of trips of the infeasible integer solution, we can add the following solution cut to the surrogate relaxed master problem:

$$\sum_{r\in\bar{\mathcal{R}}} \theta_r \leq |\bar{\mathcal{R}}| - 1 \tag{27}$$

As mentioned in article (Dall'Olio & Kolisch, 2023), this solution cut is only related to the already generated trips and does not affect the reduced cost for newly generated trips. Therefore, it does not bring any changes to the structure of the sub-problem. To prevent trips in the set $\bar{\mathcal{R}}$ from being re-added to the master problem, we directly prohibit the addition of these trips to the master problem when the sub-problem generates trips in set $\bar{\mathcal{R}}$. After adding a solution cut (referenced as (27)), to ensure that no potential optimal solutions are excluded, we need to incorporate trips of other staff combinations from trips in the set $\bar{\mathcal{R}}$ into the master problem.

5.2. Branching strategy

We explore the B&B tree according to a best-first strategy, *i.e.*, we prioritize the unexplored nodes whose parent node has the best lower bound. In our BPC algorithm, a classical branching strategy on arcs is adopted. Let $\{\tilde{\theta}_r\}$ be the optimal linear relaxation solution of the LMP at a certain B&B node. Then, the total flow over arc (i,j) can be calculated by $\tilde{x}_{ij} = \sum \beta_{ijr}\tilde{\theta}_r$. If there exists any arc (i,j) whose flow $\tilde{x}_{ij}$ is fractional, then we branch on the value of $x_{x^*y^*}$, where arc $(x^*, y^*) \in A$ corresponds to the value $\tilde{x}_{x^*y^*}$ closest to 0.5. As such, two child nodes are produced in which the left node forbids the traverse of arc (x^*, y^*) while the right node enforces the traverse of arc (x^*, y^*). As for the left node, the arc (x^*, y^*) are removed from the graph when solving the pricing problem, and all the trips traveling through arc (x^*, y^*) are neglected when solving the LRMP. In contrast, as for the right node, the arcs grouped in set $A' = \{(x^*, j), \forall j \in N, j \neq y^*\} \cup \{(i, y^*), \forall i \in N, i \neq x^*\}$ are removed from the graph when solving the pricing problem, and all the trips traveling through any arc in A' are neglected when solving the LRMP.

6. Experiments

In our numerical experiments, we implemented our algorithm using Java programming language and utilized ILOG CPLEX Optimizer 12.8.0 as the optimization solver. The experiments were performed on a machine with an Intel(R) Xeon(R) Platinum 8163HB CPU, operating at 2.5 GHz, and equipped with 16 GB of memory. The machine was running the Windows 10 operating system, and we set the time limit for each run of the BPC for the set-covering model and CPLEX for the arc-flow model to 7,200 s for all instances. In this section, we first introduce the instance generation and parameter setting for our experiments. We then examine the performance of our proposed BPC algorithm. Next, we evaluate the value of simultaneous scheduling of vehicle routes and staff. Lastly, we conduct a sensitivity analysis to evaluate the impact of variations in the number of vehicles and staff on the non-emergency patient transportation system. The instance data and detailed experimental results for the tested instances can be found on GitHub (https://github.com/for-lab/MTVRPTW-MSM).

6.1. Instance generation and parameter setting

In this section, based on Solomon instances, we introduce the benchmark instances which are generated by combining the instance generation procedures of Hernandez et al. (2016) and Zhang et al. (2017). The short planning horizon and tight time windows of type 1 Solomon instances prevent the vehicle from making multiple trips, and therefore we only consider type 2 Solomon instances. For the type 2 Solomon instances with three groups of instances (C, R, and RC), we generate instances with n customers by considering the first n customers in the original Solomon instance. For lunchtime window, we divide the time horizon T into 5 equal periods and set $e'_0 = 2/5T$, $l'_0 = 3/5T$ and $L = T/10$. The seat reservation, a_i, required by patient $i \in N$, is generated randomly from the set $\{1, 2, 3\}$. Each instance involves at least one and at most three types of staff. Instances with only one type of staff involve staff at level 3. Instances with two types of staff involve staff at levels 2 and 3. Instances with three types of staff involve staff at levels 1, 2, and 3. Staff at level 1 have the highest skills, while staff at level 3 have the lowest skills. The cost of a trip for each level of staff is set to $f_1 = 50$, $f_2 = 45$, and $f_3 = 40$, respectively. The number of

Table 1
Performance comparison between CPLEX and BPC.

Instance	CPLEX					BPC							
	UB_c	LB_c	$Time_c$	Gap_c	$\#Node_c$	UB_b	LB_b	LB_{bc}	$Time_b$	Gap_{cp}	Gap_b	Gap_{bc}	$\#Node_b$
C201	675.3	675.3	442.4	0.0	74 341	675.3	641.7	655.3	1.2	0.0	3.0	5.0	6
C202	609.5	576.7	7200.1	5.4	75 731	609.5	570.0	589.6	93.5	0.0	3.3	6.5	37
C203	583.6	583.6	2551.8	0.0	40 516	583.6	561.1	575.4	58.7	0.0	1.4	3.9	79
C204	536.4	536.4	5802.4	0.0	207 396	536.4	521.0	533.4	113.0	0.0	0.6	2.9	23
C205	659.8	659.8	47.9	0.0	10 143	659.8	659.8	659.8	0.1	0.0	0.0	0.0	1
C206	624.4	601.7	7200.1	3.6	170 803	624.4	584.9	599.2	49.8	0.0	4.0	6.3	241
C207	589.5	577.4	7200.1	2.1	2 367 423	589.5	560.6	582.5	46.8	0.0	1.2	4.9	41
C208	572.7	572.7	1110.9	0.0	1 294 959	572.7	554.8	568.1	11.7	0.0	0.8	3.1	17
R201	719.1	719.1	793.0	0.0	22 115	719.1	714.1	715.1	0.5	0.0	0.6	0.7	3
R202	652.9	630.9	7200.1	3.4	461 340	652.9	651.1	652.9	2.4	0.0	0.0	0.3	1
R203	637.5	600.0	7201.3	5.9	111 668	637.5	633.6	637.5	10.7	0.0	0.0	0.6	1
R204	675.5	625.9	7201.4	7.3	335 414	675.5	646.4	657.5	121.8	0.0	2.7	4.3	223
R205	625.3	625.3	303.1	0.0	41 543	625.3	625.3	625.3	0.4	0.0	0.0	0.0	1
R206	620.9	602.8	7200.9	2.9	517 324	620.9	599.4	607.1	200.2	0.0	2.2	3.5	65
R207	615.0	593.5	7200.9	3.5	312 758	615.0	588.8	602.2	105.9	0.0	2.1	4.3	57
R208	590.1	590.1	455.7	0.0	24 599	590.1	575.7	590.1	10.6	0.0	0.0	2.4	1
R209	679.4	679.4	2563.3	0.0	437 421	679.4	677.5	679.4	5.1	0.0	0.0	0.3	3
R210	615.0	615.0	1615.0	0.0	31 324	615.0	615.0	615.0	5.7	0.0	0.0	0.0	1
R211	590.1	590.1	3685.3	0.0	582 830	590.1	579.8	590.1	18.1	0.0	0.0	1.7	1
RC201	713.2	623.2	7200.7	12.6	328 142	713.2	677.4	693.7	10.1	0.0	2.7	5.0	31
RC202	595.9	478.5	7244.2	19.7	1 038 862	595.9	578.8	595.9	22.2	0.0	0.0	2.9	1
RC203	–	–	–	–	–	588.4	542.5	574.1	360.1	–	2.4	7.8	1294
RC204	559.4	462.7	7208.3	17.3	6 132 752	559.4	537.7	559.4	48.6	0.0	0.0	3.9	1
RC205	714.0	560.2	7203.2	21.5	14 201 230	709.0	695.4	703.9	105.7	0.7	0.7	1.9	31
RC206	671.2	637.8	7200.4	5.0	14 092 283	671.2	633.9	666.6	5.7	0.0	0.7	5.6	11
RC207	462.2	385.0	7200.3	16.7	37 601 778	462.2	443.1	462.2	3.6	0.0	0.0	4.1	1
RC208	507.5	421.2	7209.6	17.0	13 762 462	507.5	507.5	507.5	9.5	0.0	0.0	0.0	1

staff members at level 3 required by patient $i \in N$, d_i^3, is randomly generated from the set $\{1, 2, 3\}$. If an instance involves at least two types of staff, and the requirement for the number of staff members at level 3 is at least 2 ($d_i^3 \geq 2$), then the number of staff members with a higher skill level required by patient $i \in N$ is randomly generated from the set $\{0, 1\}$. Based on our experimental experience, we set the parameter t_{min} to the smaller of two values: half the duration of any trip and 10 for all instances. The vehicle capacity is set to $Q = 12$. When conducting our numerical tests, we used the staff-based inequalities for all the B&B nodes, whereas we used subset row inequalities exclusively at B&B nodes with depths of no more than 3, because this method is more time-consuming.

6.2. Algorithm performance

In this subsection, we assess the performance of the proposed BPC algorithm in several aspects. Firstly, we compare its performance to the widely-used CPLEX optimization solver by solving small-sized instances. Next, we test its performance on larger-scale instances as well as varying numbers of skill levels of staff. Finally, we evaluate the effectiveness of the proposed staff-based inequalities. Due to limited space, we only present the statistical data of the experimental results, except for the first set of experiments.

6.2.1. Performance comparison between CPLEX and BPC

In this section, we compare the performance of the proposed BPC algorithm solving the set-covering formulation with the widely-used CPLEX optimization solver solving the arc-flow formulation. We execute both algorithms on the same set of instances, each with 16 customers, 2 vehicles each with a capacity of 12, and 4 staff members. Among the staff members, one has a skill level of 2, while the other three have a lower level of 3. The feasible upper bound on the number of trips k_{UB} in the arc-flow formulation is set to 6. We report their performance in terms of solution quality and computational efficiency illustrated in Table 1. Where columns UB_c, $\#Node_c$ and UB_b, $\#Node_b$ represent the upper bounds obtained and the number of nodes explored by the BPC algorithm and the CPLEX solver, respectively. Columns LB_c, Gap_c and $Time_c$ denote the lower bound when the CPLEX solver terminates, the difference between the upper and lower bounds (given by $(UB - LB)/UB * 100$), and the termination time, respectively. Columns LB_b and LB_{bc} reflect the lower bounds at the root node of the BPC tree without and with valid inequalities, respectively. Column $Time_b$ records the termination time of the BPC algorithm. Columns Gap_{cp}, Gap_b, and Gap_{bc} represent the gap between UB_c and UB_b, the gap between UB_b and LB_b and the gap between UB_b and LB_{bc}, respectively. The dash indicates that CPLEX reported an out-of-memory error during the solving process and was unable to obtain the solution.

The results presented in Table 1 clearly demonstrate that the BPC algorithm proposed in this study outperforms CPLEX in terms of both solution quality and computational efficiency. The data shows that BPC is capable of solving all instances to optimality within 52.7 seconds, while CPLEX can only solve 11 out of 27 instances to optimality within an average time of 1761.0 seconds. In fact, CPLEX is unable to solve the remaining 16 instances to optimality even after running for two hours. It is worth noting that although CPLEX can find optimal solutions for 14 out of the 16 instances, it cannot verify their optimality due to its inefficiency in deriving tight lower bounds. Additionally, CPLEX has to explore a large number of B&B nodes, with an average of 3,626,045, while BPC only explores an average of 80 nodes. Interestingly, the root node of the BPC tree reaches optimality in 11 out of 27 instances. For the remaining 16 instances, the root node of the BPC tree provides significantly tighter valid lower bounds and narrower optimality gaps in less computational time than CPLEX. The computational time is 52.7 seconds for all instances compared to CPLEX's 1761.0 seconds for 11 optimal solved instances.

6.2.2. Performance of BPC algorithm on large-sized instances

In this subsection, we first test the proposed BPC algorithm on large-sized instances. Specifically, we test three sets of instances with 25, 40, and 50 customers, respectively. The first set of instances is associated with 2 vehicles and 5 staff members, where 1 staff member has skill level 2 and 4 members have skill level 3. The second set of instances is associated with 4 vehicles and 8 staff members, where 2 members have skill level 2 and 6 members have skill level 3. The third set of instances is associated with 4 vehicles and 10 staff members, where 3 staff members have skill level 2 and 7 members have skill

Table 2
Performance of BPC on large-size instances.

Cus	Gro	Ins	Sol	Root						Total								
				G_{lp}	T_b	G_{clp}	T_{bc}	SB_r	SR_r	T_t	SB_t	SR_t	Node	T_m	T_o	T_r	Ch_{c1}	Ch_{c2}
25	C	8	8	3.2	8.1	0.7	26.5	3.1	48.8	65.6	3.6	52.5	7.5	2.9	17.0	3.6	0.4	0.0
	R	11	11	2.2	15.9	0.5	28.1	1.4	17.0	201.8	2.7	21.7	24.6	2.3	49.4	7.8	0.6	0.0
	RC	8	8	2.3	24.4	0.6	41.2	2.5	20.0	1379.6	5.4	20.9	163.5	19.9	401.1	198.7	0.6	0.3
	All	27	27	2.5	16.1	0.6	31.5	2.2	27.3	510.4	3.8	30.6	60.7	7.7	144.0	63.1	0.6	0.1
40	C	8	6	1.9	18.0	0.5	58.1	8.3	76.8	2436.2	170.7	84.3	739.8	402.2	957.6	104.9	2.0	39.0
	R	11	8	1.5	85.0	0.5	178.1	2.3	40.3	1808.2	40.4	48.0	312.8	29.5	678.2	24.8	0.0	2.1
	RC	8	7	2.1	75.2	0.2	182.8	4.7	41.7	901.3	6.3	48.6	11.1	8.5	386.5	20.0	0.7	0.0
	All	27	21	1.8	62.6	0.4	145.4	4.8	51.2	1685.3	66.2	58.6	334.2	129.0	660.8	46.1	0.8	12.0
50	C	8	4	1.3	53.4	0.2	222.3	12.5	73.3	1519.0	31.5	82.3	58.5	70.2	625.8	64.7	0.0	0.0
	R	11	1	1.3	18.0	0.6	87.5	3.0	37.0	2163.4	42.0	41.0	87.0	34.0	685.6	21.1	0.0	9.0
	RC	8	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	All	27	5	1.3	46.3	0.3	195.4	10.6	66.0	1647.9	33.6	74.0	64.2	63.0	637.7	56.0	0.0	1.8

Table 3
Performance of valid inequalities.

Class	Gro	Ins	Sol	Root				Total								
				G_{clp}	T_{bc}	k/SB	SR	T_t	k/SB	SR	Node	T_m	T_o	T_r	Ch_{c1}	Ch_{c2}
No cut	C	8	7	3.1	3.2	0.0	0.0	641.6	0.0	0.0	517.3	30.7	135.3	72.9	0.0	0.0
	R	11	8	1.9	13.1	0.0	0.0	1202.3	0.0	0.0	272.3	10.9	250.6	40.9	0.0	0.0
	RC	8	6	1.8	12.0	0.0	0.0	2710.6	0.0	0.0	408.0	20.2	513.8	830.4	1.3	0.5
	All	27	21	2.3	9.5	0.0	0.0	1446.4	0.0	0.0	392.7	20.1	287.3	277.2	0.4	0.1
SR	C	8	7	1.0	13.8	0.0	47.3	132.3	0.0	60.5	17.3	9.0	387.9	29.7	0.0	0.0
	R	11	8	0.9	32.2	0.0	24.3	1062.4	0.0	29.3	285.8	11.1	303.2	25.1	0.0	0.0
	RC	8	6	0.6	27.0	0.0	23.7	2080.9	0.0	25.8	143.8	15.0	1054.4	500.8	0.7	0.0
	All	27	21	0.8	24.6	0.0	31.0	1043.3	0.0	37.6	162.7	11.6	554.0	169.2	0.2	0.0
SB	C	8	8	2.2	12.9	2.5	0.0	334.1	44.4	0.0	200.5	17.1	61.8	63.4	0.5	0.0
	R	11	11	1.4	17.6	1.6	0.0	568.1	20.7	0.0	219.9	10.2	130.5	39.8	0.1	0.0
	RC	8	7	0.9	23.6	2.9	0.0	920.1	11.7	0.0	95.0	6.0	217.7	309.7	0.1	0.0
	All	27	26	1.5	17.8	2.2	0.0	590.9	25.6	0.0	180.3	11.2	132.8	119.7	0.2	0.0
k-path	C	8	7	2.8	3.9	0.9	0.0	641.6	54.3	0.0	474.4	32.2	132.9	80.6	0.4	0.0
	R	11	8	1.8	13.8	0.3	0.0	1202.3	22.5	0.0	250.1	11.6	277.3	37.4	0.0	0.0
	RC	8	6	1.7	13.2	0.8	0.0	2710.6	25.3	0.0	277.2	14.9	517.8	658.9	0.5	0.2
	All	27	21	2.1	10.3	0.6	0.0	1446.4	33.9	0.0	332.6	19.4	297.9	229.4	0.3	0.0
SR+SB	C	8	8	0.7	26.5	3.1	48.8	65.6	3.6	52.5	7.5	2.9	17.0	3.6	0.4	0.0
	R	11	11	0.5	28.1	1.4	17.0	201.8	2.7	21.7	24.6	2.3	49.4	7.8	0.6	0.0
	RC	8	8	0.6	41.2	2.5	20.0	1379.6	5.4	20.9	163.5	19.9	401.1	198.7	0.6	0.3
	All	27	27	0.6	31.5	2.2	27.3	510.4	3.8	30.6	60.7	7.7	144.0	63.1	0.6	0.1
SR+k	C	8	7	1.0	11.4	1.6	44.6	125.4	1.9	54.9	12.7	6.6	38.3	4.4	0.0	0.0
	R	11	8	0.9	30.2	0.4	13.9	1043.1	4.8	20.0	79.0	4.9	235.1	16.1	0.3	0.3
	RC	8	6	0.6	26.0	0.7	15.2	1771.7	3.0	18.8	94.8	11.8	365.5	892.6	1.2	8.2
	All	27	21	0.8	22.8	0.9	24.5	945.4	3.3	31.3	61.4	7.5	206.8	262.6	0.4	2.4

level 3. The results are summarized in Table 2. For each instance set, the number of customers of the instances is labeled by ‘Cus’. The statistical results of the three instance groups (labeled by ‘Gro’) are categorized by their geographic information, *i.e.*, the geographical data are randomly generated in group ‘R’ and clustered in group ‘C’, and a mix of random and clustered structures in group ‘RC’. The number of instances and the number of instances solved to optimality are denoted by ‘Ins’ and ‘Sol’, respectively. All columns, except for the first four columns, give the average values over the optimally solved instances. The computational results associated with the root node and the complete run of the BPC algorithm are provided in blocks ‘Root’ and ‘Total’, respectively. The columns labeled as ‘G_{lp}’ and ‘G_{clp}’ under the ‘Root’ block indicate the percentage difference between the objective of the optimal solution obtained from the linear programming relaxation and that of the optimal integer solution, without and with considering valid inequalities. The columns labeled ‘T_b’ and ‘T_{bc}’ under the ‘Root’ block represent the time spent on solving the linear relaxation using CG, without and with incorporating valid inequalities. The numbers of staff-based and subset row inequalities added at the root node are reported in the columns labeled as ‘SB_r’ and ‘SR_r’ under block ‘Root’. The results of columns ‘T_t’, ‘SB_t’, and ‘SR_t’ under block ‘Total’ are similar to those under block ‘Root’ but for the complete run of the BPC algorithm in obtaining optimal integer solutions. The column ‘Node’ displays the number of nodes explored by our BPC algorithm in the branch-and-bound process. The columns ‘T_m’, ‘T_o’ and ‘T_r’ represent the computational time spent on the phases of solving the master problem, searching for ‘optional trips’, and identifying a trip with minimal reduced cost, respectively. And the ‘Ch_{c1}’ and ‘Ch_{c2}’ reflect the number of times the infeasible integer solutions occur in the branch-and-bound tree, caused by feasibility check case 3.5 and case 1 respectively.

It can be seen from Table 2 that the proposed algorithm is capable of solving all instances with 25 customers to optimality, with an average computation time of 510.4 seconds. However, as the number of customers increases to 40 and 50, only 21 out of 27 and 5 out of 27 instances, respectively, can be solved to optimality. This indicates that the computational complexity increases sharply with the number of customers, particularly when the number reaches 50. The reasons for the difficulty in solving some instances include: On one hand, as the size of the problem instances increases, the difficulty of solving the algorithm also increases. On the other hand, when some instances fail during case 3.5 or case 1, it is often necessary to halve the $t_{\min}$ parameter or introduce a solution cut to eliminate the solution at that node. At this point, the column generation algorithm must be called again to solve the node. However, upon re-solving, the optimal solution at that node may not be an integer, necessitating further branching. This increases the number of nodes in the branch-and-bound tree.

6.2.3. Performance of CPLEX and BPC algorithm on a varying number of skill levels among staff

In this subsection, we evaluate the performance of CPLEX and BPC algorithm on a varying number of skill levels among staff. For detailed information, please refer to Appendix C.

6.2.4. Performance of staff-based inequalities

In this subsection, we evaluate the importance of the proposed staff-based inequalities. As previously mentioned, the staff-based inequalities can be considered as an improvement over traditional k-path inequalities when the additional seats occupied by staff required by customers in a trip are taken into account. To assess the performance, we compare the results obtained from six different cases presented in Table 3. These cases include the BP algorithm without any added cuts (labeled as 'No cut'), the BPC algorithm with only subset row inequalities (labeled as 'SR'), the BPC algorithm with only staff-based inequalities (labeled as 'SB'), the BPC algorithm with only k-path inequalities (labeled as 'k-path'), the BPC algorithm with both subset row and staff-based inequalities (labeled as 'SR+SB'), and the BPC algorithm with both subset row and k-path inequalities (labeled as 'SR+k'). We test a set of instances with 25 customers, 2 vehicles, and 5 staff members, where 1 member has skill level 2 and 4 members have skill level 3. The number of staff-based or k-path inequalities added at the root node is recorded by the column 'k/SB' under block 'Root'. The definition of column 'k/SB' under block 'Total' are similar to those under block 'Root' but for the complete run of the BPC algorithm to obtain optimal integer solutions. The definitions of the remaining columns in Table 3 are the same as those introduced in Section 6.2.2.

The results demonstrate that staff-based inequalities outperform traditional k-path inequalities in computational effectiveness and efficiency. The BPC algorithm with staff-based inequalities achieves optimality in 26 out of 27 instances, with an average computation time of 590.9 seconds and exploration of 180.3 B&B nodes. On the other hand, the BPC algorithm with traditional k-path inequalities only solves 21 out of 27 instances to optimality, with a 144.8% increase in computational time and an 84.5% increase in the number of B&B nodes. When considering the BPC algorithm with subset row inequalities as a benchmark, the addition of k-path inequalities barely improves the algorithm's performance. However, incorporating staff-based inequalities significantly enhances the algorithm's performance. Specifically, it solves all the instances to optimality, consuming only 510.4 seconds and exploring 60.7 B&B nodes. Furthermore, the results demonstrate that the BPC algorithm with both staff-based and subset row inequalities exhibits superior performance when compared to the BP algorithm without valid inequalities. The BP algorithm is competent to solve only 21 out of 27 instances, with an average computation time of 1446.4 seconds and exploration of 392.7 B&B nodes.

6.3. Computational results for real data

In this subsection, we evaluate the performance of the BPC algorithm by testing it on real-world instances. For detailed information, please refer to Appendix D.

6.4. Value of jointly optimizing vehicle routing and staff scheduling

In this subsection, we evaluate the benefits of jointly optimizing vehicle routing and staff scheduling (JO), by comparing JO to a sequential optimization strategy (SO), *i.e.*, planning the vehicle routes first and then scheduling the staff. For detailed information, please refer to Appendix E.

6.5. Sensitivity analyses

Since the MTVRPTW-MSM under investigation is a new variant of the MTVRPTW, we conduct sensitivity analyses on key parameters, including the number of vehicles and the number of staff members. For detailed information, please refer to Appendix F.

7. Conclusions

This study addresses the planning challenges that arise in non-emergency patient transportation services in the healthcare industry by investigating a practical variant of the MTVRPTW with multi-skilled manpower requirements (for ambulance operatives at different skill/qualification levels). In this research problem, the VRPTW for transportation dominates the total operational time and cost structure, whereas workforce planning is secondary. This is different from the multi-skilled workforce scheduling and routing problem (Dall'Olio & Kolisch, 2023) which can be formulated as a VRPTW but the service time at customer sites dominate, and the cost associated with transport is not a major concern. Two mathematical formulations are developed for this innovative research problem: an arc flow model and a trip-based set-covering model. We propose an effective BPC algorithm for solving the set-covering model. In the BPC framework, we develop a novel two-phase CG algorithm to solve the subproblem and address the multi-trip characteristics of the proposed problem. In the first phase, we propose a tailored labeling algorithm to obtain non-dominated labels. Each label corresponds to numerous trips with a considerable number of possible departure times. In the second phase, we propose a strategy for identifying the trip with the minimum reduced cost for each label. We also propose staff-based inequalities to improve the lower bound of the model.

Extensive numerical experiments based on Solomon's instances are conducted to evaluate the performance of the proposed algorithm. The results show that the proposed BPC algorithm significantly outperforms CPLEX in terms of both solution quality and computational efficiency. Our algorithm can successfully solve instances with up to 50 patient transportation requests, four vehicles (ambulances), and 10 staff members (ambulance operatives) to optimality within two hours. In addition, the proposed staff-based inequalities outperform traditional k-path inequalities. Furthermore, we demonstrate the value of jointly optimizing vehicle routing and staff planning, which can result in significant cost savings of up to 20.3%. According to the sensitivity analyses, given a fleet with a certain number of vehicles (ambulances), as the utilization of vehicles increases, the total operating cost initially decreases and then reaches saturation, whereas the computational complexity of the solution algorithm remains unchanged. Similarly, given the staff with a certain number of ambulance operatives, as the utilization of staff increases, the total operating cost initially decreases, but then remains unchanged at a saturation point. As the staff utilization decreases, the computational complexity of the algorithm increases. instance, approximately 47% of patients were late for a hospital appointment.

CRediT authorship contribution statement

Nan Huang: Data curation, Formal analysis, Methodology, Resources, Software, Writing – original draft, Writing – review & editing. **Hu Qin:** Funding acquisition, Resources, Supervision. **Yuquan Du:** Data curation, Writing – original draft. **Li Wang:** Data curation, Formal analysis, Resources, Writing – original draft, Writing – review & editing.

Acknowledgements

The work described in this paper was supported by a grant from National Natural Science Foundation of China (No. 71971090, 71821001). The authors would like to thank the anonymous reviewers and associate editor for their helpful suggestions and very thorough review of the paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2024.06.025.

References

Alvarez, A., & Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, *83*, 1–12.

Archetti, C., Bouchard, M., & Desaulniers, G. (2011). Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. *Transportation Science*, *45*(3), 285–298.

Azi, N., Gendreau, M., & Potvin, J. Y. (2007). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, *178*(3), 755–766.

Azi, N., Gendreau, M., & Potvin, J. Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, *202*(3), 756–763.

Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, *59*(5), 1269–1283.

Bard, J. F. (2004). Staff scheduling in high volume service facilities with downgrading. *Iie Transactions*, *36*(10), 985–997.

Bard, J. F., & Purnomo, H. W. (2005). A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Economic Planning Sciences*, *39*(3), 193–213.

Billionnet, A. (1999). Integer programming to schedule a hierarchical workforce with variable demands. *European Journal of Operational Research*, *114*(1), 105–114.

Cappanera, P., Requejo, C., & Scutellà, M. G. (2020). Temporal constraints and device management for the skill VRP: Mathematical model and lower bounding techniques. *Computers & Operations Research*, *124*, Article 105054.

Cordeau, J. F., Stojković, G., Soumis, F., & Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, *35*(4), 375–388.

Dall'Olio, G., & Kolisch, R. (2023). Formation and routing of worker teams for airport ground handling operations: A branch-and-price-and-check approach. *Transportation Science*.

De La Vega, J., Munari, P., & Morabito, R. (2019). Robust optimization for the vehicle routing problem with multiple deliverymen. *Central European Journal of Operations Research*, *27*(4), 905–936.

De La Vega, J., Munari, P., & Morabito, R. (2020). Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, *124*, Article 105062.

Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, *58*(1), 179–192.

Desaulniers, G., Desrosiers, J., & Solomon, M. M. (2006). *Column generation*: *vol. 5*, Springer Science & Business Media.

Dunbar, M., Froyland, G., & Wu, C. L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Science*, *46*(2), 204–216.

Dunbar, M., Froyland, G., & Wu, C. L. (2014). An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers & Operations Research*, *45*, 68–86.

England NHS (2021). Improving non-emergency patient transport services: Report of the non-emergency patient transport review. https://www.england.nhs.uk/publication/improving-non-emergency-patient-transport-services/.

Hernandez, F., Feillet, D., Giroudeau, R., & Naud, O. (2014). A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4or*, *12*(3), 235–259.

Hernandez, F., Feillet, D., Giroudeau, R., & Naud, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, *249*(2), 551–559.

Huang, N., Li, J., Zhu, W., & Qin, H. (2021). The multi-trip vehicle routing problem with time windows and unloading queue at depot. *Transportation Research Part E: Logistics and Transportation Review*, *152*, 102370.

Huang, N., Qin, H., Xu, G., & Wan, F. (2024). An enhanced exact algorithm for the multi-trip vehicle routing problem with time windows and capacitated unloading station. *Computers & Operations Research*, *168*, 106688.

Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, *56*(2), 497–511.

Khodabandeh, P., Kayvanfar, V., Rafiee, M., & Werner, F. (2021). A bi-objective home health care routing and scheduling model with considering nurse downgrading costs. *International Journal of Environmental Research and Public Health*, *18*(3), 900.

Li, Y., Lim, A., & Rodrigues, B. (2005). Manpower allocation with time windows and job-teaming constraints. *Naval Research Logistics*, *52*(4), 302–311.

Lim, A., Zhang, Z., & Qin, H. (2017). Pickup and delivery service with manpower planning in Hong Kong public hospitals. *Transportation Science*, *51*(2), 688–705.

Liu, M., Luo, Z., & Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research, Part B (Methodological)*, *81*, 267–288.

Luo, Z., Liu, M., & Lim, A. (2019). A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science*, *53*(1), 113–130.

Luo, Z., Qin, H., Zhu, W., & Lim, A. (2016). Branch-and-price-and-cut for the manpower routing problem with synchronization constraints. *Naval Research Logistics*, *63*(2), 138–171.

Macedo, R., Alves, C., de Carvalho, J. V., Clautiaux, F., & Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, *214*(3), 536–545.

Martinelli, R., Pecin, D., & Poggi, M. (2014). Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, *239*(1), 102–111.

Mercier, A., Cordeau, J.-F., & Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, *32*(6), 1451–1476.

Munari, P., & Morabito, R. (2018). A branch-price-and-cut algorithm for the vehicle routing problem with time windows and multiple deliverymen. *Top*, *26*(3), 437–464.

Paradiso, R., Roberti, R., Laganá, D., & Dullaert, W. (2020). An exact solution framework for multitrip vehicle-routing problems with time windows. *Operations Research*, *68*(1), 180–198.

Pureza, V., Morabito, R., & Reimann, M. (2012). Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *European Journal of Operational Research*, *218*(3), 636–647.

Qiu, H., Wang, D., Yin, Y., Cheng, T. E., & Wang, Y. (2022). An exact solution method for home health care scheduling with synchronized services. *Naval Research Logistics*, *69*(5), 715–733.

SCAS (2017). South Central Ambulance Service. https://int.sussex.ics.nhs.uk/clinical_documents/non-emergency-patient-transport-service-nepts-in-sussex-2/.

Su, X., Xu, G., Huang, N., & Qin, H. (2023). A branch-and-price-and-cut for the manpower allocation and vehicle routing problem with staff qualifications and time windows. *Advanced Engineering Informatics*, *57*, Article 102093.

Weide, O., Ryan, D., & Ehrgott, M. (2010). An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, *37*(5), 833–844.

Yang, Y. (2022). An exact price-cut-and-enumerate method for the capacitated multitrip vehicle routing problem with time windows. *Transportation Science*.

Zhang, Z., Qin, H., Wang, K., He, H., & Liu, T. (2017). Manpower allocation and vehicle routing problem in non-emergency ambulance transfer service. *Transportation Research Part E: Logistics and Transportation Review*, *106*, 45–59.