Exact and heuristic methods for optimization in distributed logistics

Schrotenboer, Albert

DOI:
10.33612/diss.112911958

Chapter 2

A branch-and-price-and-cut algorithm for resource constrained pickup and delivery problems

Abstract. *We study a multi-commodity multi-period resource constrained pickup-and-delivery problem inspired by the short-term planning of maintenance services at offshore wind farms. In order to begin a maintenance service, different types of relatively scarce servicemen need to be delivered (transported) to the service locations. We develop resource-exceeding route (RER) inequalities, which are inspired by knapsack cover inequalities, in order to model the scarcity of servicemen. In addition to a traditional separation approach, we present a column-dependent constraints approach so as to include the RER inequalities in the mathematical formulation. An alternative pricing strategy is developed to correctly include the column-dependent constraints. The resulting approach is broadly applicable to any routing problem that involves a set of scarce resources. We present a branch-and-price-and-cut algorithm to compare both approaches that include RER inequalities. The branch-and-price-and-cut algorithm relies on efficiently solving a new variant of the Elementary Resource Constrained Shortest Path Problem, using a tailored pulse algorithm developed specifically to solve it. Computational experiments show that the RER inequalities significantly tighten the root node relaxations. The column-dependent constraints approach searches then the branch and bound tree more effectively and appears to be competitive with the traditional separation procedure. Both approaches are able to solve instances of up to 92 nodes over 21 periods to optimality.*

This chapter is based on Schrotenboer, Ursavas, and Vis (2019a):
Schrotenboer AH, Ursavas E, Vis IFA, 2019a *A branch-and-price-and-cut algorithm for solving resource constrained pickup and delivery problems. Transportation Science* 53(4):1001–1022

2.1 Introduction

The short-term planning of maintenance services at geographically scattered locations is a frequently encountered optimization problem in maintenance service logistics. At the core of these optimization problems lies the daily planning and routing of a scarce set of resources in order to peform maintenance services. We will study such a problem at offshore wind farms, a fairly new area of optimization that has received the attention of researchers lately, and refer to the problem as the Multi-period Service Planning and Routing Problem (MSPRP) (Dai, Stålhane, and Utne 2015, Stålhane, Hvattum, and Skaar 2015). Typical for these optimization problems is the restricted availability of differently skilled servicemen, which may be viewed as a scarce set of resources needed for performing maintenance services. In this paper, we present effective valid inequalities to model the scarcity of resources, and, based on those inequalities, we develop an exact solution approach that relies on a new variant of column-dependent constraints. The resulting approach is broadly applicable to routing problems that consume such a scarce set of resources (e.g., the MSPRP). In addition, we will develop the first sophisticated exact solution approach for short-term maintenance planning at offshore wind farms. This enables us to study a setting without predefined planning restrictions, as opposed to current approaches in the literature.

In the MSPRP, a service is begun if the right amount of spare parts and the right number of differently skilled servicemen are delivered to the service location. After completion of the service, the servicemen need to be picked up again to be delivered to their next-scheduled service. These delivery and pickup tasks are performed by a heterogeneous fleet consisting of vessels and helicopters, each capacitated for the total weight of the spare parts and the number of servicemen. We will refer to this fleet by using the general term “vehicle”. Note that the vehicles are not dedicated to a single serviceman, whereas, in onshore operations, vehicles are often “owned” by the servicemen: see, for example, Zamorano and Stolletz (2017); and Chen, Thomas, and Hewitt (2016). In addition, we let the travel costs and travel time be arbitrarily given for each vehicle and period, allowing the modeling of a wide variety of application dependent characteristics in a unified manner. For example, the different cost structures of corrective and preventive maintenance services (Stålhane, Hvattum, and Skaar 2015), as well as the influence of weather conditions on the maximum allowed travel time in each period (Kerkhove and Vanhoucke 2017), can be characterized in this way.

We assume that every service can be started and completed within a single period. Vehicles are allowed to continue with the remaining delivery and pickup tasks following the delivery of the servicemen at a service location, but they remain responsible for the pickup of the servicemen delivered. Service times and designated maximum daily working hours of the servicemen need to be respected. The number of available servicemen is restricted in every period, that is, it can be considered as a resource whose total consumption among different vehicle routes must respect its limited availability.

We model the MSPRP as a multi-commodity, multi-period pickup-and-delivery problem. We aim to develop cost-minimizing routes such that spare parts and servicemen are picked up and delivered between service locations, for each vehicle in each period, assuring the start and completion of all maintenance services. We develop a new mathematical formulation based on Resource-exceeding Route (RER) inequalities that model the scarcity of servicemen (resources). The RER inequalities are included by means of column-dependent constraints. We will prove that the new formulation is stronger than a standard set-covering formulation for a broad class of instance characteristics. Its use is not restricted to the MSPRP; it is broadly applicable for routing problems that involve a scarce set of restricted resources. In order to test the competitiveness of the column-dependent constraints approach, a traditional separation procedure for including RER inequalities is presented as well.

To include the column-dependent constraints in a branch-and-price (or branch-and-price-and-cut) framework, an alternative, and optimal, pricing strategy is proposed. The general performance of the branch-and-price-and-cut algorithm relies on efficiently solving pricing problems that are obtained by decomposing the problem for each vehicle and period. The pricing problems are a new variant of the Elementary Resource Constrained Shortest Path Problem (Irnich and Desaulniers 2005), and are solved by a tailored pulse algorithm (Lozano, Duque, and Medaglia 2015). We propose efficient lower bounds that are exploited in the pulse algorithm. Finally, the strength of the branch-and-price-and-cut algorithm is shown by solving a case for maintenance service logistics at offshore wind farms, which is a newly created situation and one practically inspired.

The remainder of this section will review the relevant literature and highlight the paper's contributions. First, we discuss recent developments in algorithms to solve mathematical formulations with column-dependent constraints. Second, we discuss some closely related pickup-and-delivery problems and their most recent exact solution approaches. Finally, we review recent work on short-term planning for maintenance services at offshore wind farms.

The first contribution of this paper is the formulation and use of a new variant of column-dependent constraints, that is, the RER inequalities. Column-dependent constraints are constraints that are generated for every column or variable (Feillet et al. 2010). Its use in column generation applications expressly reveals these difficulties; the number of constraints grows with the number of columns, thereby causing identification issues when generating new columns. We are able to overcome this difficulty through the development of an alternative pricing strategy.

A framework for handling column-dependent constraints with a decomposition into two subproblems is developed in Muter, Birbil, and Bülbül (2013). This work has recently been extended to an arbitrary number of subproblems in Maher (2015). Unlike these studies, the structure that we study exhibits interaction between the variables generated in the different subproblems. By exploiting this specific problem structure, we are able to develop a general and optimal pricing strategy that is broadly applicable to resource-constrained routing problems.

Our second contribution is the development of a branch-and-price-and-cut algorithm for the MSPRP, a problem that exhibits a combination of multiple traditional pickup and delivery structures. Pickup and delivery problems, as reviewed in Berbeglia et al. (2007); Parragh, Doerner, and Hartl (2008); and Battarra, Cordeau, and Iori (2014), involve finding cost-minimizing routes to satisfy transportation requests between pickup and delivery locations. A particular class of pickup-and-delivery problem is the vehicle routing problem with pickups and deliveries, where a one-to-one relation between pickup and delivery nodes exists (Dumas, Desrosiers, and Soumis 1991, Savelsbergh and Sol 1995). In the MSPRP, a one-to-one delivery and pickup structure exists between nodes that represent the start and completion of a service. State-of-the-art solution approaches are developed in Ropke, Cordeau, and Laporte (2007), and Ropke and Cordeau (2009). The former introduced many valid inequalities and tested the performance in a branch-and-cut algorithm, where the latter developed a branch-and-price-and-cut algorithm. Another exact algorithm, based on dual ascent heuristics and a cut-and-column generation procedure, is developed by Baldacci, Bartolini, and Mingozzi (2011). The most recent work is presented by Gschwind et al. (2018), where the authors present new dominance rules that allow for a bidirectional labeling algorithm.

Another class of pickup and delivery problem exhibits a many-to-many pickup and delivery structure, that is, locations demand or supply one or multiple commodities and picked-up supply may be used to satisfy demand. This pickup and delivery structure is encountered in the MSPRP between different maintenance services, that is, the picked-up servicemen can then be used to begin a new maintenance service. A branch-and-cut approach for a single vehicle is presented in Hernández-Pérez and Salazar-González (2004). This has recently been extended to multiple commodities in Hernández-Pérez and Salazar-González (2014). Looking into the multiple types of servicemen and the heterogeneous fleet being present in the MSPRP, another observation is made. If we consider a single servicemen type and a homogeneous fleet, the MSPRP can be categorized as a pickup and delivery problem with maximum travel time (Subramanian and Cabral 2008, Polat et al. 2015).

Summarizing, the MSPRP mixes a one-to-one pickup and delivery structure (between nodes representing the same service) with a many-to-many pickup and delivery structure (between different services). In addition, respecting the service times of the maintenance services yields so-called delayed precedence constraints between the delivery and pickup of servicemen, that is, the earliest possible departure time at the pickup location depends on the arrival time at the corresponding delivery location. This pickup and delivery relationship is typical for offshore applications, see, for example, Irawan et al. (2017).

The mix of traditional pickup and delivery structures leads to a new variant of the Elementary Resource Constrained Shortest Path Problem as a pricing problem. Because both travel costs and servicemen costs are being minimized, no efficient dominance criteria can be developed. We therefore propose an efficient pulse algorithm (Lozano, Duque, and Medaglia 2015) to solve the pricing problems, since that approach does not depend on dominance criteria. It relies on calculating lower bounds instead, which appears effective for the MSPRP.

The paper's third contribution is the development of a branch-and-price-and-cut algorithm in the area of offshore wind maintenance service logistics, which is the first sophisticated exact solution method in the setting we are studying. In the MSPRP we are studying a general, new setting of a single large offshore wind farm that is operated from a single depot without predefined planning restrictions. Some related studies exist, however; offshore wind farm maintenance service logistics was first encountered in Dai, Stålhane, and Utne (2015), and a follow up was presented by Stålhane, Hvattum, and Skaar (2015). They proposed a set covering formulation with a heuristic labeling algorithm to solve the pricing problems, but restricted it to a single period, whereas the MSPRP is situated in a multi-period setting. A first attempt to exactly solve realistically sized instances is presented by Irawan et al. (2017). They propose a route-enumeration strategy to solve up to eight maintenance services for three wind farms operated from two depots in a three-period planning horizon. The restriction that a route only contain services from a single wind farm reduces the complexity of the problem drastically, only at the expense of a slight increase in complexity due to the inclusion of multiple depots. In point of fact, only with a heuristic approach were they able to solve instances of up to 12 services per wind farm. Inherently, such a route enumeration approach is deemed impossible for the MSPRP, since we have no restrictions on the planning of services. With the branch-and-price-and-cut algorithm we propose optimal solutions for instances of up to 45 services in a single wind farm.

More generally speaking, offshore wind maintenance service logistics, and thus the MSPRP, falls into a particular stream of the technician routing and scheduling literature (Pillac, Gueret, and Medaglia 2013, Pillac, Guéret, and Medaglia 2018). It entails the design of routes and schedules for technicians such that a set of services is performed in a cost-minimizing way. The main difference between the MSPRP and onshore applications (Paraskevopoulos et al. 2017) is how vehicles are operated; vehicles in offshore applications are flexibly deployed to satisfy transportation requests throughout the time horizon, whereas vehicles are typically assigned upfront to servicemen in onshore applications. Recently, technician's ability to become more experienced in an activity is discussed by Chen, Thomas, and Hewitt (2016). This is extended to the stochastic case in which activities are uncertain (Chen, Thomas, and Hewitt 2017). Another recent work discusses the combined maintenance and routing problem (López-Santana et al. 2016), in which machines deteriorate stochastically over time. We acknowledge that those innovations in onshore applications may be of relevance for offshore wind maintenance service logistics. However, since offshore operations differ structurally from onshore operations, and we present the first sophisticated approach for solving a large-scale maintenance service logistics problem in offshore wind farms, we leave it for further research to assess the impact of incorporating the earlier described onshore innovations.

The remainder of this paper is as follows. We give a mathematical description of the MSPRP and, by means of a Danzig-Wolfe reformulation, a set covering formulation in Section 2.2. In Section 2.3, we describe valid inequalities for the MSPRP. In particular, we discuss the RER inequalities, the new formulation based on column-dependent constraints, and the accompanying optimal pricing strategy. Sections 2.4 and 2.5 discuss the pulse algorithm developed for solving the pricing problems and the overall structure of the branch-and-price-and-cut algorithm, respectively. Computational experiments showing the performance of the branch-and-price-and-cut algorithm and the impact of the valid inequalities are presented in Section 2.6. We conclude the paper in Section 2.7.

2.2 Problem description

In this section, we will provide a mathematical representation of the Multi-period Service Planning and Routing Problem (MSPRP) in the form of a Mixed Integer Program (MIP). After this, a standard set-covering formulation will be presented.

2.2.1 Mixed integer programming formulation

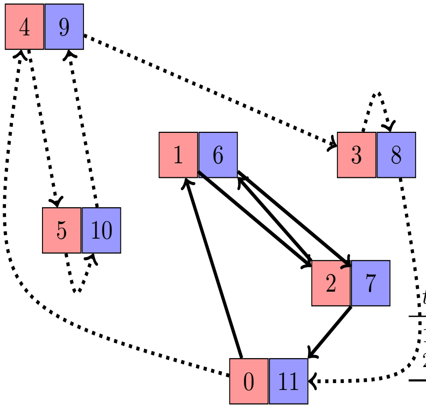
Let $G = (N, A)$ be a directed graph with a set of nodes N and a set of arcs $A = \{(i, j) \mid i, j \in N, i \neq j\}$. The node set N consists of delivery nodes $N_d = \{1, \dots, n\}$, pickup nodes $N_p = \{n+1, \dots 2n\}$, and the origin and destination depot $\{0, 2n+1\}$. Every delivery node i has a corresponding pickup node $n+i$, which represent the start and the completion of service i, respectively.

Let $\mathcal{T} = \{1, \dots, T\}$ be the given time horizon in which every $t \in \mathcal{T}$ represents a single period. We consider a set of different type of servicemen $\mathcal{L} = \{1, \dots, L\}$. The demand for service-person type $\ell \in \mathcal{L}$ at node $i \in N$ is given by $Q_{i\ell} \geq 0$, and it holds that $Q_{n+i,\ell} = -Q_{i\ell}$. The number of available servicemen is restricted; there are $\tilde{Q}_\ell$ servicemen of type ℓ available in each period. The fixed costs of using a service-person of type ℓ equals $\tilde{c}_\ell$ for each period. A precedence constraint exists between node i and $n+i$; node $n+i$ can be visited, at the earliest, s_i time after visiting node i, where s_i denotes the duration of service i. The weight of the demanded spare parts is given by $\widehat{Q}_i > 0$ for each $i \in N_d$.

A heterogeneous set of capacitated vehicles $\mathcal{K} = \{1, \dots, K\}$ is available to deliver and pickup the required servicemen in each period. As is typical for offshore operations, we assume that all vehicles are different. For each arc $(i, j) \in A$, the costs incurred of traversing it with vehicle k in period t equals c_{ij}^{kt} and the corresponding travel time equals t_{ij}^{kt}. Maintenance costs are included in the travel costs c_{ij}^{kt}, and we do not pose any restrictions on its modeling. This flexibility has two aims. First, it allows us to make a distinction between preventive maintenance tasks, in which maintenance costs are constant over the periods, and corrective maintenance tasks, in which maintenance costs increase over the periods. Second, we can model the relative urgency of the maintenance services, that is, higher costs reflect a greater urgency to perform a particular maintenance service. Both aims are easily achieved by introducing exogenously given penalty costs for not performing a maintenance service in a particular period (which could be zero).

Each vehicle $k \in K$ is capacitated in the total number of servicemen $\bar{Q}_k^1$ and the total amount of spare parts $\bar{Q}_k^2$ it can transport. The maximum travel time of vehicle k in period t equals ω_{kt}. This reflects the restrictions on performing offshore maintenance services due to weather conditions.

Let x_{ij}^{kt} be a binary decision variable that equals 1 if if vehicle k traverses arc (i, j) in period t, and 0 otherwise. Let $q_{i\ell}^{kt}$ be a nonnegative decision variable that indicates the number of servicemen of type $\ell \in \mathcal{L}$ in vehicle k in period t upon leaving node i. Finally, let z_i^{kt} be a nonnegative decision variable that equals the time at which vehicle k leaves node i at period t.



(a) *Network and route visualization*

$i, n+i$	s_i	Q_{i1}	Q_{i2}
1, 6	2.0	3	2
2, 7	3.0	1	2
3, 8	1.5	2	1
4, 9	1.5	3	1
5, 10	3.5	1	2

(b) *Overview of instance characteristics*

t	route	q_{01}^{1t}	q_{02}^{1t}	Dur.
1	$0-1-2-6-7-11$	4	4	4.5
2	$0-4-5-10-9-3-8-11$	4	3	7.5

(c) *Route characteristics*

Figure 2.1: An illustrative example of the MSPRP.

To highlight the complexity of the MSPRP, an illustrative example is provided in Figure 2.1.

Example 2.1. Let $n = 5$, $T = 2, K = 1$, and $L = 2$. Spare parts demand $\widehat{Q}_i$ equals 0 and servicemen availability $\tilde{Q}_\ell = 4$ for all $\ell \in \{1, 2\}$. Maximum travel time ω_{1t} equals 6 and 12 for $t = 1$ and $t = 2$, respectively. In Figure 2.1(b), characteristics of the services are provided, and in Figure 2.1(c), a feasible solution is depicted. "Dur." indicates the travel time of the corresponding route. We assume that all the arcs' travel times equal 0.5, except for the edges between corresponding delivery and pickup locations, those are assumed to take zero time in this example. Some calculations are as follows: 1) Regarding the duration of the route in Period 1. Let $a := t_{01} + t_{12} + \max\{t_{01} + s_1, t_{01} + t_{12} + t_{26}\}$ be the earliest possible departure from Node 6. The earliest possible departure from Node 7 is then $\max\{a + t_{67}, t_{01} + t_{12} + s_2\} := b$. The route duration is then equal to $b + t_{7,11}$, which equals 4.5 in this example. 2) The servicemen use in Period 2 is the sum of the servicemen used for Services 4 and 5, since Service 3 is supplied with servicemen that become available after having finished Services 4 and 5. 3). Note that a single route in Period 2 $(0-1-2-6-7-4-5-10-9-3-8-11)$ is a feasible solution as well, as its duration is less than 12 and the servicemen use equals the maximum of both individual routes. $\triangleleft$

The following MIP models the MSPRP.

$$\min \quad \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{(i,j)\in A} c_{ij}^{kt} x_{ij}^{kt} + \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} \sum_{\ell\in\mathcal{L}} q_{0\ell}^{kt} \tilde{c}_\ell \tag{2.1}$$

subject to

$$\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\sum_{j:(i,j)\in A} x_{ij}^{kt} = 1 \qquad \forall i \in N_d, \tag{2.2}$$

$$\sum_{j:(i,j)\in A} x_{ij}^{kt} - \sum_{j:(j,i)\in A} x_{ji}^{kt} = 0 \qquad \forall\ i \in N_d \cup N_p, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.3}$$

$$\sum_{j:(0,j)\in A} x_{0j}^{kt} = 1 \qquad \forall\ k \in K, t \in \mathcal{T}, \tag{2.4}$$

$$\sum_{j:(j,2n+1)\in A} x_{j,2n+1}^{kt} = 1 \qquad \forall\ k \in K, t \in \mathcal{T}, \tag{2.5}$$

$$\sum_{j:(i,j)\in A} x_{ji}^{kt} - \sum_{j:(n+i,j)\in A} x_{n+i,j}^{kt} = 0 \qquad \forall\ i \in N_d,\ k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.6}$$

$$t_{ij}^{kt} x_{ij}^{kt} - M(1 - x_{ij}^{kt}) \leq z_j^{kt} - z_i^{kt} \qquad \forall\ (i,j) \in A,\ k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.7}$$

$$z_i^{kt} + s_i \leq z_{i+n}^{kt} \qquad \forall\ i \in N_d,\ k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.8}$$

$$Q_{j\ell} x_{ij}^{kt} - M(1 - x_{ij}^{kt}) \leq q_{i\ell}^{kt} - q_{j\ell}^{kt} \qquad \forall\ (i,j) \in A,\ \ell \in \mathcal{L},\ k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.9}$$

$$\max\{0, -Q_{i\ell}\} \leq q_{i\ell}^{kt} \qquad \forall\ i \in N_d \cup N_p,\ \ell \in \mathcal{L}, k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.10}$$

$$\sum_{\ell\in\mathcal{L}} q_{j\ell}^{kt} \leq \min\{\bar{Q}_k^1, \bar{Q}_k^1 + \sum_{\ell\in\mathcal{L}} Q_{i\ell}\} \qquad \forall\ j \in\in N_d \cup N_p, k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.11}$$

$$\sum_{(i,j)\in A: j\in N_d} x_{ij}^{kt} \widehat{Q}_j \leq \bar{Q}_k^2 \qquad \forall\ k \in \mathcal{K},\ t \in \mathcal{T} \tag{2.12}$$

$$z_{2n+1}^{kt} \leq \omega_{kt} \qquad \forall\ k \in \mathcal{K}, t \in \mathcal{T}, \tag{2.13}$$

$$\sum_{k\in\mathcal{K}} q_{0\ell}^{kt} \leq \tilde{Q}_\ell \qquad \forall\ \ell \in \mathcal{L},\ t \in \mathcal{T}, \tag{2.14}$$

$$x_{ij}^{kt} \in \{0, 1\} \qquad \forall\ (i,j) \in A,\ k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.15}$$

$$q_{i\ell}^{kt} \geq 0 \qquad \forall\ i \in N,\ \ell \in \mathcal{L},\ k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.16}$$

$$z_i^{kt} \geq 0 \qquad \forall\ i \in N,\ k \in \mathcal{K},\ t \in \mathcal{T}. \tag{2.17}$$

Objective (2.1) minimizes the costs of traveling and for servicemen usage. The travel costs may include maintenance costs or penalty costs. Constraints (2.2) ensure that every node is visited only once and constraints (2.3) are the traditional flow conservation constraints. Constraints (2.4) and (2.5) ensure that every route starts and end at the origin and destination depot, respectively. The vehicle that delivers the servicemen must also pickup the servicemen, as denoted by Constraints (2.6).

Constraints (2.7) and (2.8) model travel and service times, respectively. In (2.7), M denotes a big enough number so that the constraints are redundant if they need to be. A valid value of M is ω_{kt}. Constraints (2.9) model the servicemen demand at every node. With Constraints (2.10) and (2.11) we strengthen the lower bound and upper bound of $q_{j\ell}^{kt}$, respectively. The maximum capacity for spare parts is respected due to Constraints (2.12). Finally, Constraints (2.13) limit the maximum travel time of a vehicle and Constraints (2.14) ensure feasibility with respect to the limited availability of servicemen.

The MIP formulation exhibits an interesting structure. It is decomposable for every vehicle $k \in \mathcal{K}$ and period $t \in \mathcal{T}$. The constraints that link the decisions among the (k,t)-subproblems are Constraints (2.2) and (2.14). We, therefore, apply a Dantzig-Wolfe reformulation resulting into a set-covering formulation as presented in the following section.

2.2.2 Set covering formulation

Let $\mathcal{R}$ be the set of all feasible routes that can be constructed in the MSPRP. A route's costs and feasibility may differ between vehicles and periods, since vehicles are heterogeneous and arc costs, as well as maximum travel times, differ among periods. Therefore, let $\mathcal{R} = \cup_{k\in\mathcal{K},t\in\mathcal{T}}\mathcal{R}_{kt}$, where $\mathcal{R}_{kt}$ denotes the set of feasible routes of vehicle k in period t. For notational convenience, let $\mathcal{R}_k = \cup_{t\in\mathcal{T}}\mathcal{R}_{kt}$ and $\mathcal{R}_t = \cup_{k\in\mathcal{K}}\mathcal{R}_{kt}$. For each route $r \in \mathcal{R}_{kt}$, let y_r be a binary decision variable that equals 1 if route r is chosen and 0 otherwise. In addition, let c_r be the corresponding costs, β_i^r be the number of times node $i \in N_d$ is visited, and γ_ℓ^r be the number of servicemen of type $\ell \in \mathcal{L}$ used by route $r \in \mathcal{R}_{kt}$.

A set-covering formulation is then given by:

$$\min \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_{kt}} c_r y_r \tag{2.18}$$

$$\text{s.t.} \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_{kt}} y_r \beta_i^r \geq 1 \qquad \forall\, i \in N_d, \tag{2.19}$$

$$\sum_{r\in\mathcal{R}_{kt}} y_r \leq 1 \qquad \forall\, k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.20}$$

$$\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_{kt}} y_r \gamma_\ell^r \leq \tilde{Q}_\ell \qquad \forall\, \ell \in \mathcal{L},\ t \in \mathcal{T}, \tag{2.21}$$

$$y_r \in \{0,1\} \qquad \forall r \in \mathcal{R}_{kt}, k \in \mathcal{K},\ t \in \mathcal{T}. \tag{2.22}$$

The objective (2.18) minimizes the costs for using the selected routes from each subset

$\mathcal{R}_{kt}$. Constraints (2.19) ensure that every node is visited at least once. Constraints (2.20) ensure that every vehicle in every period is used at most once, which is necessary due to the heterogeneity of vehicles and periods. Constraints (2.21) ensure that the maximum number of servicemen used in every period does not exceed the servicemen availability. We will refer to the model described by equations (2.18)-(2.22) as the Integer Programming Master (IPM) problem. Its linear relaxation, obtained by replacing Constraints (2.22) with $y_r \geq 0$, is referred to as the Linear Programming Master (LPM) problem.

Due to the exponential size of $\mathcal{R}$, solutions to LPM are usually obtained by column generation (Barnhart et al. 1998). To that extent, consider restricted route sets $\overline{\mathcal{R}}_{kt} \subset \mathcal{R}_{kt}$. Notice that by a Dantzig-Wolfe decomposition we arrive at $K \cdot T$ subproblems, that is, for vehicle k and period t we obtain the (k,t)-pricing problem. We iteratively solve LPM subject to $\overline{\mathcal{R}}_{kt}$ and generate new routes for each $\overline{\mathcal{R}}_{kt}$ by solving the (k,t)-pricing problems. Model LPM is solved if no route of negative reduced cost can be found for any (k,t)-pricing problem. Then, a dual optimal solution is found and by strong duality it is a primal optimal solution as well.

To formulate the (k,t)-pricing problems, let μ_i, λ_{kt}, and π_ℓ^t be dual variables corresponding to Constraints (2.19) - (2.21), respectively. Let d_{ij}^{kt} be the costs of traversing arc (i,j) in some (k,t)-pricing problem. We define

$$d_{ij}^{kt} = \begin{cases} c_{ij}^{kt} - \mu_j & \text{if } j \in N_d, \\ c_{ij}^{kt} & \text{otherwise.} \end{cases} \tag{2.23}$$

Similarly, let $\tilde{d}_\ell^t = \tilde{c}_\ell - \pi_\ell^t$ be the reduced servicemen costs in an arbitrary (k,t)-pricing problem. Then the (k,t)-pricing problems are given by

$$\min_{r \in \mathcal{R}_{kt}} \left\{ \sum_{(i,j) \in A} d_{ij}^{kt} r_{ij} + \sum_{\ell \in \mathcal{L}} \tilde{d}_\ell^t \gamma_\ell^r - \lambda_{kt} \right\}, \forall\, k \in \mathcal{K},\ t \in \mathcal{T}, \tag{2.24}$$

where $r_{ij} = 1$ for all arcs (i,j), $j \neq n+i$, which are used by path $r \in \mathcal{R}_{kt}$, and is 0 otherwise.

An illustrative example of a route in an arbitrary (k,t) pricing problem is presented in Figure 2.2.

Example 2.2. Let $K = T = L = 1$ and let $n = 5$. Consider three Services 1, 2 and 3 that demand 2, 2, and 3 servicemen, respectively. As observed from Figure 2.2, only $\tilde{c}_1 - \pi_1^1$ reduced servicemen costs are incurred when visiting Service 3 (instead of $3\tilde{c}_1 - 3\pi_1^1$) since 2 servicemen just become available after visiting node 7. Furthermore,

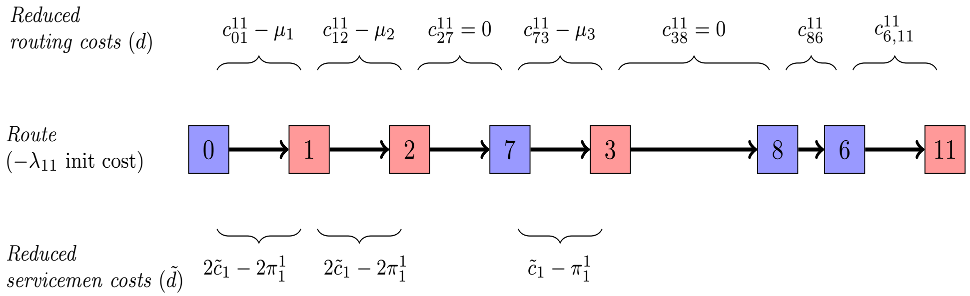


Figure 2.2: An illustrative example of a route in a (k, t) pricing problem.

note that the traveling costs between corresponding delivery and pickup nodes equals zero, and observe that only reduced servicemen costs are considered when visiting a delivery node. $\triangleleft$

The (k, t)-pricing problems are new variants of the Elementary Resource Constrained Shortest Path Problem. In line with Irnich and Desaulniers (2005), it can be categorized as a multi-commodity resource constrained shortest path problem with delayed paired precedence constraints, a problem that, to the best of the authors' knowledge, has not been solved before. The delayed paired precedence constraints refer to the one-to-one relationship between pickup and delivery nodes in combination with the corresponding service time constraint. For simplicity, we refer to this new variant of the Elementary Resource Constrained Shortest Path Problem by using the general term Pricing Problem (PP). The main complicating factor that appears in (2.24) are the servicemen costs $\tilde{d}^t_\ell$, as is indicated in Figure 2.2 as well. These cannot be incorporated directly into individual edge costs, henceforth leading to algorithmic difficulties, as will be explained in detail in Section 4.

2.3 Valid inequalities

Model IPM, presented in Section 2.2.2, is a set covering formulation as encountered in many branch-and-price (and branch-and-price-and-cut) approaches. It is, although valid for solving the MSPRP, relatively weak due to the inclusion of knapsack-type Constraints (2.21). These constraints cannot be taken into account in the pricing problems directly, thereby reducing the overall efficiency, since it weakens the LP relaxation. In this section, we will develop *Resource-exceeding Route* (RER) inequalities, which are a specialized form of knapsack cover inequalities (Gu, Nemhauser, and Savelsbergh 1999,?). They are applied in branch-and-price algorithms for the generalized assignment problem (Savelsbergh 1997) and multicommodity flow problems (Barnhart, Hane, and Vance 2000), for instance. The RER inequalities very efficiently restrict the number of servicemen that routes can use, since this is a resource "consumed" by vehicles whose its availability at the depot is restricted. What differs here from the current applications of cover inequalities is that the sequence of visits within a tour may change the resource consumption. This causes difficulties during the pricing of routes. The inclusion of multiple resource types complicates the construction of RER inequalities further, which can be observed in the remainder of this section.

We present two approaches for including RER inequalities that are tractable in pricing problems. In the first approach, we develop a new formulation of the MSPRP in which we replace Constraints (2.21) with column-dependent Constraints (Muter, Birbil, and Bülbül 2013). These constraints are generated when the restricted route set (as used in column generation) is enlarged. Its applicability is not restricted to this case only: The proposed reformulation and corresponding solution approach are applicable for any linear program being solved with column generation in which a system of constraints as described by (2.21) is present. The approach is easy to understand and it provides interesting theoretical insights into the inclusion of valid inequalities. In order to test its competitiveness, we propose a second approach that separates the RER inequalities by a traditional separation procedure, which is called upon dynamically during the branch-and-price-and-cut algorithm. So, where the first approach adds the RER inequalities *while* generating new routes, the second approach adds RER inequalities *after* generating new routes.

In the following, we will first develop the alternative formulation for the MSPRP by using column-dependent constraints. The column-dependent constraints are problem defining, and a valid formulation of the MSPRP is obtained by replacing the knapsack-type Constraints (2.21). We refer to this alternative formulation as the Alternative Integer Programming Master Problem (AIPM). We analyze the strength of the formulation and prove that for special cases its LP relaxation is stronger than LPM. We present an alternative pricing strategy and prove its optimality. After this, a description of a traditional separation procedure of RER inequalities is given, in order to be able to judge the computational performance of the column-dependent constraints approach. We will end this section with some well-known valid inequalities that are included in our exact algorithm as well.

2.3.1 Column-dependent constraints approach

By exploiting the notion of *resource-exceeding routes*, that is, routes that cannot simultaneously enter a feasible integer solution due to their servicemen use, we will replace the relatively weak Constraints (2.21) by a set of column-dependent constraints that restricts the use of resource-exceeding routes. Consider the following example. It illustrates the concept of resource-exceeding routes and of the corresponding valid inequalities.

Example 2.3. Let $L, T = 1$ and $\tilde{Q}_\ell = 6$. Consider routes $r_i \in \mathcal{R}_{it}, i \in \{1,2,3\}$ with $\gamma_1^{r_i} = 3$ such that $r_i = (0, i, n+i, 2n+1)$. In other words, Vehicle 1 performs Service 1, Vehicle 2 performs Service 2, and Vehicle 3 performs Service 3. Then the constraint $\sum_{i \in \{1,2,3\}} y_{r_i} \leq 2$ is valid, whereas no restriction should be put on any subset of these routes. Consider now a fourth route r_4 with $\gamma_1^{r_4} = 4$. Then there are i valid inequalities of the form $y_{r_4} + y_{r_i} \leq 1$ for all $i \in \{1,2,3\}$. $\lhd$

The valid inequalities shown in Example 1 are problem-defining for the MSPRP. A valid formulation of the MSPRP is obtained if we include those valid inequalities in IPM and leave out Constraints (2.21). In the following, we present an approach that generates such valid inequalities for any subset of routes from different vehicles, of size at least 1 and at most $K-1$. For every subset, we include column dependent constraints that restrict the maximum number of allowed routes, so that the servicemen availability is respected.

2.3.1.1 Resource-exceeding route inequalities

Let $k(r)$ and $t(r)$ be the vehicle and period index of route $r \in \mathcal{R}_{kt}$, respectively. The complement route set R_r^C of route r is defined as the set of routes from different vehicles in the same period as r, that is, $\mathcal{R}_r^C := \{\tilde{r} \in \mathcal{R} \mid t(\tilde{r}) = t(r), k(\tilde{r}) \neq k(r)\}$. We call a set of routes a *partial solution* if it contains at most a single route for each vehicle and if it does not contain a route for all the vehicles. The collection of partial solutions is formally defined as follows:

Definition 2.1 (partial solution)**.** The collection $\mathbb{S}_t$ of partial solutions is defined as

$$\mathbb{S}_t := \{S \subseteq \mathcal{R}_t \mid |S_k| \leq 1 \ \forall k \in K, 1 \leq |S| \leq K-1\}, \tag{2.25}$$

with $S_k = \{r \in S \mid k(r) = k\}$.

Now the collection of route sets $\mathbb{S}_t$ contains all partial solutions, and for each of those partial solutions we need to check how many of those routes can be present in a feasible integer solution. Therefore, we let ϕ_S be the maximum number of allowed routes from $S \in \mathbb{S}_t$ in a feasible integer solution, that is.,

$$\phi_S := \max \left\{ |v| : \sum_{v \in V \subseteq S} \gamma_\ell^v \leq \tilde{Q}_\ell \ \forall \ell \in \mathcal{L} \right\} \tag{2.26}$$

We will continue with a brief example to clarify the intuition behind ϕ_S and $\mathbb{S}_t$.

Example 2.4. Consider the setting as in example 1 with routes r_1, r_2, r_3 and r_4. Two partial solutions are given by $S_1 = \{r_1, r_4\}$ and $S_2 = \{r_1, r_2, r_4\}$. Here $\phi_{S_1} = 1$ as only r_1 or r_4 could be present in a feasible integer solution. On the other hand, $\phi_{S_2} = 2$ as r_1 and r_2 can both be present in a feasible integer solution. ◁

The concept of RER inequalities is as follows. Suppose we select a partial solution, the subset of whose routes is part of a feasible integer solution. Since it is a partial solution, some routes for other vehicles (of which no routes are contained in the selected partial solution) could be part of the optimal solution as well. We can, however, put restrictions on the use of those routes for other vehicles in combination with a partial solution, depending on ϕ_S.

To model these restrictions neatly, we introduce $\gamma_\ell^{\phi_S}$ as the minimum use of servicemen over the sets V that results in ϕ_S, see (2.26). Consider the following three special cases for the value of $\gamma_\ell^{\phi_S}$. First, for $|S| = 2$ and $\phi_S = 1$, $\gamma_\ell^{\phi_S}$ equals the minimum use of servicemen type ℓ among the routes from S. Second, for $|S| = 2$ and $\phi_S = 2$ it results in the sum of servicemen types ℓ among the routes from S. Finally, for $|S| = 3$ and $\phi_S = 2$, $\gamma_\ell^{\phi_S}$ equals the minimum sum of servicemen use among two routes of S, for each ℓ independently.

For any feasible integer solution we ensure that at most ϕ_S routes are selected from a partial solution $S \in \mathbb{S}_t$. Let the complement route sets of S be given by $R_S^C = \{r \in R \mid \forall s \in S,\ t(s) = t(r),\ k(r) \in k^C(S)\}$, where $k^C(S) = \{C \subset \mathcal{K} \mid k' \neq k(s), \forall s \in S, k' \in C\}$. It reflects all routes belonging to vehicles for which no routes are contained in the partial solution S but are in the same period as the routes of S. Then we can formally define a resource-exceeding route as:

Definition 2.2 (Resource-exceeding Route)**.** A route $r \in \mathcal{R}_{kt}$ is resource-exceeding with respect to some partial solution $S \in \mathbb{S}_t$ if it belongs to the set E_S of resource-exceeding routes with respect to S. The set E_S is defined as:

$$E_S := \left\{ \tilde{r} \in \mathcal{R}_S^C \mid \exists\ \ell \in \mathcal{L}\ : \gamma_\ell^{\tilde{r}} + \gamma_\ell^{\phi_S} > \tilde{Q}_\ell \right\} \tag{2.27}$$

Hence a route is resource-exceeding with respect to a partial solution S if it cannot be added to the partial solution without changing the value of ϕ_S.

To model RER inequalities, let $\Upsilon_{S,r} = 1$ if $r \in E_S$, and 0 otherwise. In addition, let $\Gamma_{S,r} = 1$ if there exists an $s \in S$ such that $k(s) = k(r)$ and for all $\ell \in \mathcal{L}$ it holds that $\gamma_l^s \geq \gamma_\ell^r$, and 0 otherwise. Then constraints (2.21) can be rewritten as the complete set of RER inequalities

$$\sum_{r\in\mathcal{R}_t} \Gamma_{S,r} y_r + \sum_{r\in\mathcal{R}_S^C} \Upsilon_{S,r} y_r \leq \phi_S \quad \forall\, C \in k^C(S),\ S \in \mathbb{S}_t,\ t \in \mathcal{T}. \tag{2.28}$$

We call S the constraint-generating subset of routes. We refer to RER inequalities of size s as the constraints (2.28) when $|S| = s$.

The model described by equations (2.18)-(2.20), (2.22) and (2.28) is a valid description of the MSPRP, and we refer to it as the Alternative Integer Programming Master Problem (AIPM). Its Linear Relaxation is referred to as ALPM. Since we will work with only a subset $\overline{\mathcal{R}} \subset \mathcal{R}$ in our pricing procedure, let $\overline{\mathbb{S}}_t$ denote $\mathbb{S}_t$ with respect to the restricted route set $\overline{\mathcal{R}}$. The resulting restricted integer and linear programming master problems are then denoted by RAIPM and RALPM, respectively.

An extensive comparison between the different models is given in Section 2.6. We will especially focus on the strength of the root node relaxations of models IPM and AIPM. In addition, we will investigate the effect of extending IPM with (2.28) and of extending AIPM with (2.21).

2.3.1.2 Properties of resource exceeding route inequalities

First, we show that AIPM has stronger LP relaxations then IPM, for $L = 1$. This is characterized by the following proposition.

Proposition 2.1 (Improved relaxation)**.** *Assume that $L = 1$. Let $c(ALPM)$ and $c(LPM)$ be the objective values of model ALPM and LPM, respectively. Then for any given $\overline{\mathcal{R}} \subseteq \mathcal{R}$, it holds that $c(ALPM) \geq c(LPM)$*

Proof. We show that the constraints generated by (2.21) are a subset of the constraints generated by (2.28). Without loss of generality, assume that $T = 1$, $y_r > 0$ for all $r \in \bar{\mathcal{R}}$ and that $\gamma_\ell^r \neq \gamma_\ell^{r'}$ $(r \neq r')$ for any $r, r' \in \bar{\mathcal{R}}_k$. Consider an arbitrary constraint (2.21) that is violated. It implies that there is a minimum set of routes U such that $\sum_{u\in U} y_u \gamma_u > \tilde{Q}_\ell$; otherwise, constraint (2.21) cannot be violated. We show that there will always exist a RER inequality (2.28) that is violated if the following procedure is followed. For every vehicle K whose routes are included in U, let γ_ℓ^k be the minimum servicemen use of that vehicle. Let $U' \subset U$ be the collection of routes with that minimum resource usage among the K vehicles. Then, for at least a single subset $U'' \subset U'$, an RER inequality (2.28) with generating subset of routes U'' will be violated. This follows directly from the definition of $\Upsilon(\cdot)$ and $\Gamma(\cdot)$. Hence the constraints generated by (2.21) are a subset of the constraints generated by (2.28), and therefore the objective value of ALPM is at least the objective value of LPM, that is, $c(\text{ALPM}) \geq c(\text{LPM})$. □

Hence, for a single resource (i.e., servicemen) type, model ALPM provides us with tighter LP relaxations. However, the differences between AIPM and IPM become less clear when multiple resource types are considered. The following propositions, therefore, provide insights into the differences between the models for practical scenarios.

Proposition 2.2. *Let $t \in \mathcal{T}$ be arbitrarily given and assume that $L = 1$. Let $S \subset \mathbb{S}_t$ be an arbitrary partial solution such that $\sum_{s \in S} \gamma_\ell^s > Q_\ell$. Then for any solution y with $\sum_{r \in S} y_r > \phi_S$ and $\sum_{r \in R_t} y_r \gamma_\ell^r \leq Q$ it follows that $y \notin$ ALPM while $y \in$ LPM.*

Proposition 2.3. *Let $t \in \mathcal{T}$ be arbitrarily given. Suppose that for all $r \in \mathcal{R}$ it holds that $\gamma_\ell^r = Q_\ell$ for all $\ell \in \mathcal{L}$. Then $y \in$ (LPM) implies that $y \in$ (ALPM) and therefore Conv(ALPM) = Conv(LPM).*

We briefly sketch the proofs of Propositions 2 and 3, as they follow trivially by the definitions of $\Upsilon(\cdot)$ and $\Gamma(\cdot)$. For Proposition 2, notice that $\sum_{r \in R_t} y_r \gamma_\ell^r \leq Q$ implies that constraints (2.21) are satisfied, while constraints (2.28) are not satisfied since $\sum_{s \in S} \gamma_\ell^s > Q_\ell$. For Proposition 3, all servicemen are transported by a single vehicle, and therefore constraints (2.21) and (2.28) coincide.

Propositions 2.2 and 2.3 tell us that AIPM is especially beneficial compared with IPM if the number of servicemen is restrictive but not fully utilized by single vehicles, that is, servicemen are maximally deployed but among different vehicles. In addition, both propositions made clear that for many (practical) route sets $\bar{\mathcal{R}}$, it may hold that $c(\text{ALPM}) > c(\text{LPM})$. This especially holds for the MSPRP, where a shared resource (servicemen) is completely utilized among multiple vehicles.

2.3.1.3 Pricing of resource exceeding route inequalities

Difficulties arise during the pricing step of the branch-and-price-and-cut algorithm. Normally, the added columns are found by solving the pricing problem, and we continue doing that until no new columns are found. However, when a new column is added to ALPM, new constraints must be included as well (constraints (2.28)). These constraints are, however, not yet known when the pricing problem is solved and columns might, therefore, be priced incorrectly. For general linear programs, this could lead to incorrectly priced columns and non-terminating pricing procedures. In the next subsection, we will describe an alternative pricing strategy that will provide optimal solutions to ALPM and is guaranteed to terminate.

Let d_{ij}^{kt}, r_{ij} and λ_{kt} be as defined in Section 2.2.2 and let ψ_S be the dual costs corresponding to constraints (2.28). For readability, let $\mathbb{S} = \cup_{t\in\mathcal{T}}\mathbb{S}_t$ and let $\overline{\mathbb{S}}$ be the set $\mathbb{S}$ with respect to the restricted route set $\overline{\mathcal{R}}$. The (reduced) servicemen costs in the (k,t)-pricing problem corresponding to some route r equals

$$\overline{d}_r = \sum_{\ell\in\mathcal{L}} \tilde{c}_\ell \gamma_\ell^r - \sum_{S'\in A(S), S\in\mathbb{S}_t} (\Gamma_{S,r} + \Upsilon_{S,r})\psi_{S'} - \sum_{S'\in B(S), S\in\mathbb{S}_t} (\Gamma_{S,r} + \Upsilon_{S,r})\psi_{S'}, \quad (2.29)$$

where $A(S)$ is the set of constraints (2.28) generated by route set S already existing when route r is being generated, and $B(S)$ is the set of constraints (2.28) being generated by all generating subsets S that include r. That is, the constraints in $A(S)$ can be automatically priced , since r can only enter as a resource exceeding route in those inequalities, whereas the constraints in $B(S)$ do not exist at the moment of pricing, since they are generated due to generating r.

The first term of equation (2.29) consists of the primal service-person costs for the generated route r, the second term consists of the dual costs corresponding to constraints (2.28) where route r enters, and the third term consists of the dual costs corresponding to constraints (2.28) that are not yet in RALPM at the moment of generating r. The completely specified (k,t)-pricing problem then equals

$$\min_{r\in\overline{\mathcal{R}}_{kt}} \hat{c}_r := \sum_{(i,j)\in A} d_{ij}^{kt} r_{ij} + \bar{d}_r - \lambda_{kt}, \quad (2.30)$$

In (2.29), the dual values corresponding to the constraints generated by including the new columns are unknown, and therefore (2.29) cannot be determined at the moment of generating new routes. In the remaining part of this section, we will develop an alternative pricing strategy that provides us with an optimal solution to ALPM and is guaranteed to terminate as well.

For readability, let $\Delta(r) = \sum_{S'\in B(S), S\in\overline{S}_t} (\Gamma_{S,r} + \Upsilon_{S,r})\psi_{S'}$, and let $\hat{\bar{d}}_r := \bar{d}_r + \Delta(r)$. Since $\psi_{S'} \leq 0$ for all S', it follows that $\Delta(r) \leq 0$ and subsequently that $\hat{\bar{d}}_r \leq \bar{d}_r$ for all $r \in \mathcal{R}$. By replacing $\bar{d}_r$ with $\hat{\bar{d}}_r$ in (2.29), we arrive at the alternative (k,t)-th pricing problem:

$$\min_{r\in\overline{\mathcal{R}}_{kt}} \hat{\hat{c}}_r := \sum_{(i,j)\in A} d_{ij}^{kt} r_{ij} + \hat{\bar{d}}_r - \lambda_{kt}. \quad (2.31)$$

All dual values corresponding to RER inequalities are nonnegative and, consequently,

Algorithm 2.1: Column and Row Generation procedure (CRG)

```
while true do
    LP ← SolveRLPM();
    for k ∈ K do
        for t ∈ T do
            Set of columns S ← SolveAlternativePricing(k, t, LP);
            if S = empty then
                goTo line 2;
            end
        end
    end
    break;
end
```

$\hat{\hat{c}}_r \leq \hat{c}_r$. Hence the alternative pricing problem will correctly identify routes that cause dual infeasibility after being included in the restricted route set, which is needed for a correct column generation approach. The following proposition summarizes the above reasoning.

Proposition 2.4. *Let $k \in \mathcal{K}$ and $t \in \mathcal{T}$ be arbitrarily given. Consider the (k,t)-pricing problem* (2.29) *and the alternative (k,t)-pricing problem* (2.31). *Consider an arbitrarily dual solution and corresponding to this dual solution, let $\hat{R} := \{r \in \mathcal{R}_{kt} \mid \hat{c}_r < 0\}$ and $\hat{\hat{R}} := \{r \in \mathcal{R}_{kt} \mid \hat{\hat{c}}_r < 0\}$. Then $\hat{R} \subseteq \hat{\hat{R}}$.*

From this proposition, we conclude that using the alternative pricing problem will not result in any suboptimal solution of ALPM. What remains to be shown is that using (2.31) instead of (2.29) will result in a column generation procedure that terminates.

The complete column and row generation procedure (CRG) is outlined in Algorithm 2.1. It consists of iteratively solving the alternative (k,t)-th pricing problem and solving RALPM. If no negative reduced cost routes are found in some (k,t)-th pricing problem, we continue searching for negative reduced cost routes in the next (k,t)-th pricing problem. If negative reduced cost routes are found by some (k,t)-pricing problem, we add those to RALPM and generate constraints (2.28). We then solve RALPM and restart the procedure. The procedure terminates if there is not a single route of negative reduced cost for any (k,t)-pricing problem.

Proposition 2.5. *Let $\hat{c}_r$ and $\hat{\hat{c}}_r$ be the reduced costs according to* (2.29) *and* (2.31)*, respectively. Let $\hat{R} := \{r \in \mathcal{R}_{kt} \mid \hat{c}_r < 0\}$ and $\hat{\hat{R}} := \{r \in \mathcal{R}_{kt} \mid \hat{\hat{c}}_r < 0\}$. Then $\hat{\hat{R}} \backslash \hat{R} = \emptyset$.*

Proof. We show that there is no route r such that $\hat{c}_r > 0$ while $\hat{\hat{c}}_r$ is negative. Let $k \in \mathcal{K}$ and $t \in \mathcal{T}$ be arbitrarily given. There are two cases that we need to consider.

1. Consider the possibility that we generate a route r that is already included in $\overline{\mathcal{R}}_{kt}$. Now assume $\hat{\hat{c}}_r < 0$. We know that there exists a route r' already included in $\overline{\mathcal{R}}_{kt}$ before generating r. That implies that $\hat{c}_{r'} < 0$, since $\Delta(r')$ is already known. However, $\hat{c}_{r'} > 0$ since it is contained in the restricted route set before obtaining the optimal solution to RALPM. Hence, such a route r cannot be generated.

2. Assume we generate route $r \in \mathcal{R}_{kt} \backslash \overline{\mathcal{R}}_{kt}$ with $\hat{\hat{c}}_r < 0$ and $\hat{c}_r > 0$. If $\hat{\hat{c}} < 0$, this implies that route r cuts off the current dual solution, if we ignore the constraints (2.28) generated by r. However, including those constraints results in a larger dual feasible region. It automatically follows that route r still cuts off the same dual solution in the enlarged dual space. Hence $\hat{c}_r < 0$ as well. As a result, we have shown that there are no such routes r.

By Proposition 2.5 and the above results, it follows that $\hat{\hat{R}} \backslash \hat{R} = \emptyset$. □

Theorem 2.1. *The procedure (CRG) terminates and solves ALPM to optimality.*

Proof. Follows directly from Propositions 4 and 5. □

2.3.2 Separating resource exceeding route inequalities

The concept of resource-exceeding route inequalities is explained in depth in Section 2.3.1. A novel method of adding the inequalities based on column-dependent rows has been discussed above. In order to compare its computational efficiency, we present a traditional separation procedure for adding the resource-exceeding route inequalities in this section.

A resource-exceeding route inequality is determined by a subset $S \in S_t$, a set $C \subseteq K^C(S)$ and the resource consumption level ϕ_ℓ^S. Instead of taking the perspective from individual routes, we now take the perspective of a resource consumption level $u^k = (u_1, u_2, \ldots, u_L)$ of vehicle k. A resource level u^k and a subset of vehicle indices $\overline{\mathcal{K}} \subset \mathcal{K}$, where $k \in \overline{\mathcal{K}}$, defines a single resource-exceeding route inequality in the following way:

Let resource consumption levels $u^k = (u_1, \ldots, u_L)$, $k \in \overline{\mathcal{K}}$ be given. Consider the set of dummy routes $S' = \{r_k\}_{k \in \overline{\mathcal{K}}}$ such that route r_k has resource consumption level u^k. Then, resource-exceeding inequalities are defined as inequalities (2.28) for $S = S'$.

The two approaches for adding resource exceeding-route inequalities have their benefits and drawbacks. The column-dependent constraints approach does not rely on a separation procedure; it adds resource exceeding route inequalities *while* generating the routes. This may become less efficient when K becomes large, since the number of added inequalities will then quickly increase. On the other hand, it remains very efficient when the number of servicemen type L becomes large. In addition, the column-dependent constraints approach does not rely on the exact defintion of resource-exceeding, that is, the approach remains valid even if other applications require nonlinear relations to determine whether or not routes are resource-exceeding. The separation approach becomes relatively inefficient for larger L due to the increasing number of combinations of resource levels. In addition, the separation approach will not result in a problem-defining set of constraints, whereas the column dependent constraints approach will.

2.3.3 Other valid inequalities

We will continue by discussing two well-known valid inequalities that are included in our algorithm for solving the MSPRP. For a set of nodes $S \subseteq N$, let $\delta^+(S) := \{(i,j) \in A \mid i \in S, j \notin S\}$.

2.3.3.1 2-path inequalities

Since the MSPRP inherits aspects from the one-to-one pickup and delivery problem, we have included so-called 2-path inequalities that have been shown to be effective in a set-covering formulation (Ropke, Cordeau, and Laporte 2007, Ropke and Cordeau 2009). They are formulated as follows:

Let $S \subseteq (N_d \cup N_p)$ be such that it cannot be visited by a single vehicle k in some period t. Then the following inequality is valid for the MSPRP,

$$\sum_{y_r \in \delta^+(\mathcal{S})} y_r \geq 2. \tag{2.32}$$

The 2-path inequalities are separated by means of a greedy heuristic and an exact labeling algorithm, as is discussed in Ropke and Cordeau (2009). Since a 2-path inequality is a single cut on the y_r variables, corresponding dual costs can be incorporated in the pricing problem by subtracting it from the arc costs that leave the set S.

2.3.3.2 Subset-row inequalities

We include subset-row inequalities, in the form of Chvátal-Gomory rank 1 cuts on Constraints (2.19). They are defined as follows:

For any $S \subseteq N_d$ and $k \in \mathbb{N}$ such that $0 < k \leq |S|$,

$$\sum_{r \in \overline{\mathcal{R}}} \left\lfloor \frac{1}{k} \sum_{i \in S} \beta_i^r \right\rfloor y_r \leq \left\lfloor \frac{|S|}{k} \right\rfloor. \tag{2.33}$$

The dual values corresponding to the subset-row inequalities (2.33) are incorporated in the pricing problem as discussed by Jepsen et al. (2008). We adopted their separation procedure as well. Initial experiments have shown that including subset-row inequalities for $k = 2$ and $|S| = 3$ is computationally efficient, whereas inequalities for other values of k and $|S|$ are not effective and are therefore not taken into account.

2.4 Pricing problems

The previous section defined (k, t)-pricing problems for both LPM and ALPM. Since the structure of each pricing problem is similar, we describe how to solve an arbitrarily given (k, t)-pricing problem. We first motivate the choice for developing a new variant of the pulse algorithm. Then, we discuss how to incorporate the dual values into the pricing problem, especially those corresponding to the RER inequalities (2.28). After this, we present the pulse algorithm in detail and prove its correctness.

2.4.1 The pricing problem

The pricing problem is a new variant of the Elementary Resource Constrained Shortest Path Problem (ERCSPP), introduced by Desrochers (1987) and discussed in Feillet et al. (2004) and Irnich and Desaulniers (2005), for instance. It can be classified as a Multi-Commodity Elementary Resource Constrained Shortest Path Problem with Delayed Paired Precedence Constraints, which has, to the best of the authors' knowledge, not been solved before. For readability, we refer to it by the term Pricing Problem (PP). Let us briefly summarize the problem characteristics included in the PP. The multi-commodity part refers to the different types of servicemen demanded at each service. Each service has a pickup and a delivery node, where the delivery node needs to be visited before the pickup node (precedence relation), the pickup node needs to be visited if the delivery node is visited (pairing relation), and the corresponding service time needs to be respected between the delivery and pickup node ("delayed time" relation).

Traditionally, ERCSPPs are solved with labeling algorithms (Desrochers, Desrosiers, and Solomon 1992). These are dynamic programming approaches in which non-dominated partial paths are extended with nodes until the optimal solution is found. The extent to which non-dominated partial paths can be identified determines the labeling algorithm's computational efficiency. Dominance criteria are often based on the validity of the triangle inequality regarding the costs of traversing a path: It is more expensive to make a detour instead directly traversing an arc.

The nature of the MSPRP complicates the construction of efficient dominance criteria, since there are costs $\tilde{c}_\ell$ for using servicemen of type ℓ. This destroys the validity of the triangle inequality, i.e., it can be cheaper to make a detour instead of directly traversing an arc. The method proposed by Ropke and Cordeau (2009) to restore the, in their case, triangle inequality for the pickup nodes, can be used to partially restore the triangle inequality for the delivery nodes. In particular, travel costs (including possible penalty or maintenance costs) could be restructured as they depend solely on the arcs traversed, but the servicemen costs incurred could not.

A major consequence of this is that the dominance criteria developed by Ropke and Cordeau (2009) are not valid for solving our PP, and, henceforth, cannot be used for the PP. This leads to less label dominance and to a slower labeling algorithm. Therefore, we need to resort to another algorithm to solve PP efficiently. We develop a new variant of the pulse algorithm (Lozano, Duque, and Medaglia 2015) tailored for solving the PP without relying on dominance criteria. The pulse algorithm is shown to be competitive with the state-of-the-art labeling algorithm of Baldacci, Bartolini, and Mingozzi (2011) for solving the ERCSPP. The general overview of the pulse algorithm for solving the PP is given in Algorithms 2.3 and 2.4.

2.4.2 Incorporating the dual values

Recall that μ_i, λ_{kt}, and π^t_ℓ are the dual variables corresponding to constraints (2.19)-(2.21), and let their value be zero if one of the corresponding constraints is not included in the problem formulation. For example, π^t_ℓ equals zero in the PP of ALPM, since constraints (2.21) are replaced by constraints (2.28).

Let the initial costs of any partial path be $-\lambda_{kt}$. In order to incorporate the dual values corresponding to visiting nodes, we let the costs of traversing an arc $(i, j) \in A$ be d^{kt}_{ij} and let the costs for using a service-person of type ℓ be $\tilde{d}^t_\ell$, as defined in Section 2.2. Dual values corresponding to other cuts or valid inequalities (as long they correspond to visiting nodes or a set of nodes only) can be included in the travel

Algorithm 2.2: Construction of $\hat{C}$.

Data: Matrix $\hat{C}$, set of RER inequalities $\mathcal{S}$ with for each $S \in S$, corresponding $C \subset K^C(S)$, period t and vehicle k of the current (k,t)-th PP

for *RER inequality* $S \in \mathcal{S}$ *with corresponding set* $K^C(S)$ **do**
  for *route* $r \in S$ **do**
    if $k(r) = k$ **then**
      $u =$ findResourceLevel(r);
      setMatrixSameVehicle($u, \tilde{Q}_1, \ldots, \tilde{Q}_L$)
    end
  end
  if $k \in k^C(S)$ **then**
    $u =$ findResourceLevel(r);
    setMatrixOtherVehicle($u, \tilde{Q}_1, \ldots, \tilde{Q}_L$)
  end
end

costs by subtracting them from their corresponding arcs.

To incorporate the dual costs ψ of every RER inequality generated by all subsets $S \subset \mathcal{R}$, we define an $L-$dimensional array $\hat{C}$ of size $(\tilde{Q}_1+1)\times(\tilde{Q}_2+1)\times\cdots\times(\tilde{Q}_L+1)$. Entry $Q_{u_1\ldots,u_L}$ contains the dual costs corresponding to the RER inequalities generated by subsets S where a generated route r using $u_1,\ldots,u_L$ servicemen will enter.

The algorithm to construct $\hat{C}$ is given in Algorithm 2.2. Here, the function 'findResourceLevel(r)' returns the servicemen use u of the route r in the inequality generating subset S. Then, if $k(r) = k$, we subtract the dual cost for every entry $Q_{\tilde{u}_1,\ldots,\tilde{u}_L}$ for which $\tilde{u}_\ell \geq u_\ell$ for all $\ell \in \mathcal{L}$. However, if $k \in k^C(S)$ we subtract the dual costs for every entry $Q_{\tilde{u}_1,\ldots,\tilde{u}_L}$ if there exists an $\ell \in \mathcal{L}$ such that $\tilde{u}_\ell + \gamma_\ell^{\phi_S} > Q_\ell$, that is, the servicemen-use is such that it is resource exceeding with respect to S. All dual values corresponding to RER inequalities are non-positive. This leads to the following observation:

Proposition 2.6. *Let $u = (u_1,\ldots,u_L)$ be an arbitrarily given resource level. Then for any $u' = (u'_1,\ldots,u'_L)$, such that $u'_\ell \geq u_\ell$ for all $\ell \in \mathcal{L}$, it holds that $\hat{C}_{u_1,\ldots,u_L} \leq \hat{C}_{u'_1,\ldots,u'_L}$.*

Proof. Let r and r' be routes consuming u and u' resources, respectively. Then the set of constraints where r enters is a subset of the set of constraints where r' enters. Since all dual values are negative, it follows that $\hat{C}_{u_1,\ldots,u_L} \leq \hat{C}_{u'_1,\ldots,u'_L}$. □

Algorithm 2.3: pulse main

Data: Time $\underline{t}$, time step δ.
κ = createLowerBounds($\underline{t}, \delta$);
$\bar{z} = \infty$;
Set of negative reduced cost routes S;
pulse(L, $\{0\}, \bar{z}$);
return S;

Algorithm 2.4: pulse(L, n, $\bar{z}$)

Data: Label L, node n, best objective $\bar{z}$
if *feasible(L, n)* **then**
  if *checkBounds(L, n)* **then**
    Label L' = extend(L, n);
    if $d(L') + \tilde{d}(L') < \bar{z}$ **then**
      $\bar{z} = d(L') + \tilde{d}(L')$;
    end
    if $d(L') + \tilde{d}(L') < 0$ **then**
      $S = S \cup \{L'\}$
    end
    for $n' \in N$ **do**
      pulse($L', n', \bar{z}$);
    end
  end
end

2.4.3 The pulse algorithm for PP

As can be seen from Algorithms 2.3 and 2.4, the pulse algorithm uses a depth-first search for exploring the solution space. Pruning of partial paths is performed by a lower bound criterion, initialized in the "createLowerBounds" function. Then the "pulse" procedure is called upon recursively, in which the "feasible" and "checkBounds" procedures prune partial paths based on feasibility and a lower bound criterion, respectively. All procedures will be explained in detail subsequently.

We store the information of a partial path as a label L, consisting of the following elements:

- The corresponding partial path $\vec{\eta}(L)$;
- The vector of departure times $\vec{t}(L)$ corresponding to $\vec{\eta}(L)$,
- The travel costs $d(L)$;

- The servicemen costs $\tilde{d}(L)$ including the dual costs from $\hat{C}(L)$;
- The number of servicemen $\zeta_\ell(L)$ of type ℓ currently working at delivery nodes;
- The maximum number of servicemen $\bar{\zeta}_\ell(L)$ of type ℓ used so far;
- The cumulative weight $\xi(L)$ of spare parts delivered so far;
- The set of pickup nodes $\mathcal{S}(L)$ whose delivery node is visited and the pickup node is still unvisited;
- And a set of delivery nodes $\mathcal{U}(L)$ that have already been visited.

The notation $d(L)$ and $\vec{\eta}(L)$ is used to denote the costs and the partial path of label L, respectively. This notation is used consistently for referring to the elements of label L. For a partial path of size s, the vectors $\vec{\eta}(L)$ and $\vec{\eta}(t)$ are of dimension $1 \times s$. With $t(L)$ and $\eta(L)$ we denote the last element of $\vec{t}(L)$ and $\vec{\eta}(L)$, respectively.

We introduce time windows $[a_i, b_i]$ for nodes $i \in N$. For $i \in N_d$, time windows are set as $[t_{0i}, \omega - s_i - t_{i,2n+1}]$, and for $i \in N_p$ they are set as $[t_{0i} + s_i, \omega - t_{i,2n+1}]$. In each call of the 'pulse' function, we extend a label L with some node $i \in N$, resulting in a new label L' that is constructed as follows.

$$\vec{\eta}(L') = (\vec{\eta}(L), i), \tag{2.34}$$

$$\zeta_\ell(L') = \zeta_\ell(L) + Q_{i\ell} \quad \forall \ell \in \mathcal{L}, \tag{2.35}$$

$$\xi(L') = \xi(L) + Q_i, \tag{2.36}$$

$$\vec{t}(L') = (\vec{t}(L), \max\{t(L) + t^{kt}_{\eta(L),i},\ a_i\}), \tag{2.37}$$

$$\bar{\zeta}_\ell(L') = \max\{\zeta_\ell(L) + Q_{i\ell}, \bar{\zeta}_\ell(L)\} \quad \forall \ell \in \mathcal{L}, \tag{2.38}$$

$$\tilde{d}(L') = \sum_{\ell \in \mathcal{L}} \bar{\zeta}_\ell(L') \cdot \tilde{d}^t_\ell - \hat{C}_{\bar{\zeta}(L)} + \hat{C}_{\bar{\zeta}(L')}, \tag{2.39}$$

$$d(L') = d(L) + d^{kt}_{\eta(L),i}, \tag{2.40}$$

$$\mathcal{S}(L') = \begin{cases} \mathcal{S}(L) \cup \{n+i\} & \text{if } i \in N_d \\ \mathcal{S}(L) \backslash \{i\} & \text{if } i \in N_p \end{cases}, \tag{2.41}$$

$$\mathcal{U}(L') = \begin{cases} \mathcal{U}(L) \cup \{i\} & \text{if } i \in N_d \\ \mathcal{U}(L) & \text{if } i \in N_p \end{cases}. \tag{2.42}$$

A label is pruned if it appears to be infeasible or if it cannot improve the current best solution. Both pruning criteria will be discussed next.

2.4.3.1 Feasibility

In the "feasible" procedure, we consider a label L that we extend with some node i. Similarly as in Ropke and Cordeau (2009), we only need to consider nodes i that satisfy

$$i \notin U \text{ if } i \in N_d, \quad i \in S \text{ if } i \in N_p, \quad \mathcal{S} = \emptyset \text{ if } i = 2n+1. \tag{2.43}$$

These indicate that a delivery node can be visited once at most; a pickup node can only be visited if the corresponding delivery node is visited, and visiting the destination depot is only feasible if there are no pickup nodes unvisited whose corresponding delivery node is visited.

If a node i satisfies the precedence and pairing relationships (2.43), we need to check the feasibility of the resource constraints:

$$\xi(L) + Q_i \leq \bar{Q}^2, \quad \sum_{\ell \in L} \max\{\bar{\zeta}(L), \zeta_\ell(L) + Q_{il}\} \leq \bar{Q}^1,$$
$$\zeta_\ell(L) + Q_{il} \leq \tilde{Q}_\ell \;\forall \ell \in \mathcal{L}, \quad t(L) + t_{\eta(L),i} \leq b_i. \tag{2.44}$$

These model the spare part capacity as well as the servicemen capacity restrictions of the vehicle, the restricted availability of servicemen, and feasibility with respect to the time windows introduced. To conclude, the "feasible" procedure returns false if one of the conditions (2.43) or (2.44) is violated; otherwise true is returned.

2.4.3.2 Lower Bounds

The "createLowerBounds" procedure constructs a series of lower bounds κ_i^t for visiting node i at discrete time steps $t \in [\underline{t},\ \underline{t}+\delta,\ \underline{t}+2\delta+, \ldots, \omega-\delta]$ for all $i \in N_d$, where δ is a parameter determining the discrete time steps. The κ_i^t can be interpreted as a lower bound on the maximum possible gain for a path starting at i at time t. Let $\bar{z}$ denote the best solution to PP so far, that is, a valid upper bound on the optimal solution of the PP.

Consider a feasible label L with i being the last-added node to $\vec{\eta}$, and let t_i be departure time of i. The "checkBounds" procedure prunes a Label L if $d(L) + \tilde{d}(L) + \kappa_i^{\tilde{t}} > \bar{z}$, where $\tilde{t} \in [\underline{t},\ \underline{t}+\delta,\ \underline{t}+2\delta+, \ldots, \omega-\delta]$ and $\tilde{t} \leq t_i$. We choose $\tilde{t}$ as large as possible, since this will give the tightest lower bound at time t_i.

The lower bounds are calculated before solving the PP. They are based on the validity of the *pickup-triangle inequality* with respect to the incurred *travel* costs d, that is, the travel costs cannot decrease due to visiting an additional pickup location.

Initially, this property does not hold, since dual costs corresponding to valid inequalities and branching decisions may be included in the travel cost.

The following procedure restores the pickup-triangle inequality with respect to the travel costs: Assume that the dual costs of the pickup-triangle inequality breaking constraints are already incorporated in $\bar{d}_{ij}$. We search for the largest violation v_j of the pickup-triangle inequality, that is, $v_j := \max_{i,k \in N}\{\bar{d}_{ik} - (\bar{d}_{i,j+n} + \bar{d}_{j+n,k})\}$ for all $j \in N_d$. For all $j \in N_d$, we subtract v_j from $\bar{d}_{ij}$ and add v_j to $\bar{d}_{i,j+n}$ for all $i \in N$. For a proof of the correctness of this, we refer to Ropke and Cordeau (2009).

The following proposition describes how a valid lower bound is obtained.

Proposition 2.7 (Lower Bounds). *A valid lower bound κ_i^t for all $i \in N_d$, $t \in [\underline{t},\ \underline{t}+\delta,\ \underline{t}+2\delta+, \dots, \omega-\delta]$ is obtained by solving the pulse algorithm starting with label L, where L is defined as*

$$\vec{\eta}(L) = (i),\ d(L) = 0,\ \tilde{d}(L') = 0,\ \vec{t}(L) = (t),\ \xi(L) = Q_i$$

$$\zeta_\ell(L) = Q_{i\ell},\ \bar{\zeta}_\ell(L) = \tilde{Q}_\ell\ \forall \ell \in \mathcal{L}$$

$$\mathcal{S}(L) = \{n+i\},\ \mathcal{U}(L) = \{i\},$$

In addition, we assume that $\hat{C} = 0$ during the construction of κ_i^t.

Proof. Let $L^\star$ be the solution of the pulse algorithm if it starts with label L as defined above. Then the following two properties of $L^\star$ hold:

1. $\tilde{d}(L^\star) = 0$, since $\hat{C} = 0$ and $\bar{\zeta}_\ell(L^\star) = \tilde{Q}_\ell\ \forall \ell \in \mathcal{L}$. Hence $\bar{z} = d(L^\star)$ after running the pulse algorithm.

2. For any label $\tilde{L}$ starting at $\{0\}$ and ending at i, it holds that $\mathcal{S}(L) \subseteq \mathcal{S}(\tilde{L})$ and $\mathcal{U}(L) \subseteq \mathcal{S}(\tilde{L})$.

Then, let $\tilde{L}^\star$ be the solution of the pulse algorithm if we start with some label $\tilde{L}$ at time t, and let $d(\tilde{L}^\star) + \tilde{d}(\tilde{L}^\star)$ be its costs. By Property 2 and the pickup triangle inequality, it follows that $d(\tilde{L}^\star) \geq d(L^\star)$. By Property 1, and non-negativity of the servicemen costs, it follows that $d(L^\star) \geq d(\tilde{D}^\star)$.

Since $\kappa_i^t = \bar{z} = d(L^\star)$ after running the pulse algorithm starting with label L, it follows that $d(\tilde{L}) + \tilde{d}(\tilde{L}) + \kappa_i^t \leq d(\tilde{L}^\star) + \tilde{d}(\tilde{L}^\star)$ for any label $\tilde{L}$ starting at $\{0\}$ arriving at node i at time $\tilde{t} \geq t$. Hence κ_i^t is a valid lower bound for any $i \in N_d, t \in [\underline{t},\ \underline{t}+\delta,\ \underline{t}+2\delta+, \dots, \omega-\delta]$. $\square$

2.5 Branch-and-price-and-cut algorithm

We will now continue by presenting the general outline of the branch-and-price-and-cut algorithm. Recall that IPM is defined by equations (2.18)-(2.22), whereas AIPM is defined by equations (2.18)-(2.20), (2.22), and (2.28). We will elaborate on a simple heuristic for the construction of an initial route set, and we will discuss the branching and node selection strategy used.

Algorithm 2.5: Randomized search procedure

```
Data: Set of services S
randomSort(S);
s, s', s'' = CheapestFeasibleInsertion(S);
k = 1;
while k < k_max do
    s'' = s';
    S' = removeVessel(s'', i);
    s'' = CheapestFeasibleInsertion(s'', S');
    S' = removeJobs(s'', j);
    s'' = CheapestFeasibleInsertion(s'', S');
    k = k + 1;
    if accept(s'') then
        s' = s''
    end
    if s'' < s then
        s = s''
    end
end
```

2.5.1 Initial solution

Starting with a high-quality set of initial solutions may help to prune nodes of the branch-and-bound tree at an earlier stage. To that extent, we have used a small randomized search strategy inspired by the Two-stage Adaptive Large Neighbourhood Search in Schrotenboer et al. (2018a). It consists of randomly sequencing all services and inserting them by Cheapest Feasible Insertion. After this, it consists of an iterative procedure of two main steps: first, removing all services from a randomly selected number of routes and reinserting them in a random order with cheapest feasible insertion, and, second, removing some random services from the solution and reinserting them in a random order by cheapest feasible insertion. The new solution is accepted based on some simple simulated annealing criteria, see Schrotenboer et al. (2018a) for details. The general outline of this procedure is given in Algorithm 2.5.

Table 2.1: Benchmark characteristics.

Bm.	ω_t	$Q_{i\ell}$	$\tilde{Q}_\ell$	$\tilde{Q}_i$	s_i	p_i	$\hat{p}$	Vehicle	$\bar{Q}^1$	$\bar{Q}^2$	speed	cost
A	$[6,10]$	$[1,3]$	6	$[400,800]$	$[2,5]$	$[50,1000]$	0.25	1	9	2000	60	20
								2	12	3000	35	32
B	$[6,12]$	$[0,4]$	7	$[400,800]$	$[3,6]$	$[50,1000]$	0	1	9	2000	8	80
								2	12	3000	35	128
C	$[6,10]$	$[1,3]$	6	$[400,800]$	$[2,5]$	$[50,1000]$	0.25	1	8	2000	60	30
								2	8	2000	50	25
								3	10	2000	50	20

2.5.2 Branching and node selection strategy

Branching rules and node selection rules should be selected with care in column-generation applications. We apply branching inspired by the approach in Naddef and Rinaldi (2001). Before explaining the exact branching procedure, notice that we are branching on edges in the original formulation. We need to define the values of x_{ij}^{kt} for every $(i,j) \in A, k \in \mathcal{K}, t \in \mathcal{T}$. These can easily be obtained from the values of the LP solution $\bar{y}_r, r \in \overline{\mathcal{R}}$, that is, $x_{ij}^{kt} = \sum_{r \in \overline{\mathcal{R}}_{kt}} r_{ij}\bar{y}_r$, where r_{ij} equals 1 if edge (i,j) is visited in route r.

The branching rule is simple, but effective. We search for a set $S \subseteq (N_d \cup N_p)$, for every period t and vehicle k, such that $x^{kt}(\delta^+(S))$ is as fractional as possible. Here, $x^{kt}(\delta^+(S)) := \sum_{(i,j)\in\delta^+(S)} x_{ij}^{kt}$. A simple greedy procedure is used to determine suitable candidates S for every combination of k and t. Preliminary experiments have shown that often a set of two nodes is found for which it holds that $x^{kt}(\delta^+(S))$ - $\lfloor x^{kt}(\delta^+(S)) \rfloor = 0.5$.

When a suitable candidate set S, a period index t and a vehicle index k, are found, branching is performed by imposing the constraint $\sum_{r\in\bar{\mathcal{R}}_{kt}} \sum_{(i,j)\in\delta^+(S)} r_{ij} y_r \leq \lfloor x^{kt}(\delta^+(S)) \rfloor$ on one child node, and $\sum_{r\in\bar{\mathcal{R}}_{kt}} \sum_{(i,j)\in\delta^+(S)} r_{ij} y_r \geq \lceil x^{kt}(\delta^+(S)) \rceil$ on the other child node. The branching constraints impose single cuts on (A)IPM, whose corresponding dual values should be considered in the pricing problem. This is done by incorporating the dual values into the costs of traversing arcs $\delta^+(S)$ in the (k,t)-pricing problem. The node selection strategy, i.e., which nodes of the branch and bound tree to explore first, is based on a best first search strategy. Initial experiments have shown that this works well.

2.6 Computational experiments

The goal of this section is twofold. First, we will provide insights into the efficiency of RER inequalities for different sizes of the corresponding generating subsets. Recall that that the maximum size of the RER inequalities equals the number of vehicles minus one. Second, we will show that the column dependent approach (Section 3.1) is, besides being theoretically interesting, competitive with a traditional separation method (Section 3.2). Recall that model formulation AIPM inherently uses the column-dependent constraints approach, since the knapsack-type constraints (2.21) are replaced with RER inequalities (2.28). On the other hand, we are making use of the separation procedure when we include RER inequalities into model IPM. In particular, we refer to IPM if we use the traditional set-covering formulation without RER inequalities. If we include the RER inequalities in IPM, we will refer to it as IPM + RERx, where x is the size of the included RER inequalities.

All experiments are conducted on three newly created benchmark sets of practically inspired instances, described below. We implemented the branch-and-price-and-cut algorithm with the framework for constraint programming SCIP 3.2.1 (Gamrath et al. 2016) in combination with CPLEX 12.6.3 as an LP-solver. The overall program is coded in C++. All experiments are performed on a Xeon E5 2680v3 CPU (2.5 GHz) processor with 16 GB of RAM. The implementation is completely single-threaded. The maximum calculation time is set to 10800 seconds or the time that 16GB of RAM is used, whichever comes first.

The benchmark sets are constructed based on a practical setting of offshore wind maintenance service logistics off the coast of the Netherlands. The instance characteristics are in line with recent work in offshore wind maintenance service logistics (Dai, Stålhane, and Utne 2015, Irawan et al. 2017) and follow from interviews with stakeholders in maintenance service logistics for offshore wind farms in the Netherlands. Benchmarking our branch-and-price-and-cut algorithm with existing approaches and benchmarks in this area is either not possible due to the inclusion of multiple depots and of multiple distinct wind farms that led to restrictions in the planning of the maintenance services (Irawan et al. 2017), or the benchmark instances are too small (solved within a second) to provide additional insights (Dai, Stålhane, and Utne 2015). We would like to stress that the approach of Irawan et al. (2017), in which the complete solution space is enumerated, has its limits at 8 services per wind farm. The instance sets that we propose, and are able to solve, contain up to 45 services for a single wind farm.

The lower bound procedure incorporated in the pulse algorithm for solving the pricing problem is run with $\delta = 1$ and $\underline{t} = \lfloor 0.4\omega_t \rfloor$ for all $t \in \mathcal{T}$. The separation procedure for RER inequalities (model IPM) is a simple enumeration based on the current route set. It is run once every five branch and bound nodes, but only if the current branch and bound node provides the smallest lower bound on the optimal solution. The subset-row and 2-path inequalities are only included in the root node. This appears computationally to be most efficient.

2.6.1 Benchmark characteristics

We created three benchmark sets (A, B and C) that each entail different problem characteristics. A detailed description of the parameters is given in Table 2.1. Benchmark Set A resembles a practical situation with two vessels, a relatively fast and cheap but small vessel and a larger but slower and more expensive vessel. It consists of relatively short time horizons ($T \leq 10$). Benchmark Set B shares the same vessel characteristics as Benchmark A, but travel costs are increased. In addition, time horizons are larger ($T > 10$) and the number of services of the instances is larger. Finally, Benchmark Set C resembles a practical situation with three relatively small vessels with small differences regarding their capacity, travel costs, and travel speed. We would like to stress that the benchmarks are created such that vehicles will not be deployed in every period, since weather conditions are such that it is more profitable to cluster services in particular periods, instead of visiting the wind farm with each vehicle on a daily basis.

For each benchmark, the coordinates of the depot are fixed at $(0, 30)$, and the maintenance jobs are drawn in a box with lower left corner $(20, 20)$ and upper right corner $(40, 40)$. We consider three different types of servicemen, resembling servicemen with mechanical, electrical and electromechanical specialties. Their costs $\hat{c}_\ell$ equal $300, 325$ and 375, respectively. The demand for service-person type ℓ, as well as the weight of the spare parts needed for each service, is independent of other services' demands and is (uniformly) randomly drawn as specified in Table 2.1. This table also specifies the service times. In order to make a distinction between services, we consider a constant, per period, penalty p_i for not performing service i. The probability of a job having $p_i = 0$ equals $\hat{p}$. This resembles preventive maintenance tasks, while higher values of p_i resemble the relative urgency of performing a service and can be interpreted as corrective maintenance tasks.

The maximum driving time (in hours) of the vehicles is uniformly drawn as denoted by ω_t in Table 2.1. This resembles practical situations at offshore wind farms where daily working hours are limited due to weather conditions (e.g., wind speeds or fog). In

Table 2.2: Optimality gaps (%) of the root node relaxations for the inclusion of different valid inequalities.

		AIPM					IPM					dif.	
Bm.	n	none	2p	ss	full	t_{root}	none	2p	ss	full	t_{root}	full	t_{root}
A	10	1.14	1.05	0.73	0.73	0.12	2.46	2.34	2.16	2.16	0.17	65.96	1.35
	15	0.76	0.76	0.70	0.70	0.74	2.69	2.69	2.61	2.60	0.53	73.08	0.72
	20	1.13	1.13	1.02	1.02	4.09	2.65	2.62	2.42	2.42	6.95	57.41	1.70
	23	0.79	0.79	0.66	0.64	5.59	2.43	2.43	2.29	2.29	4.97	72.24	0.89
	26	1.33	1.33	1.28	1.27	13.36	3.27	3.27	3.23	3.23	8.14	60.66	0.61
	29	1.19	1.19	1.18	1.16	11.95	2.63	2.63	2.58	2.57	21.46	54.89	1.80
	32	1.16	1.16	1.14	1.12	30.56	2.96	2.94	2.95	2.93	48.75	61.70	1.59
	35	0.98	0.98	0.95	0.96	53.46	2.74	2.74	2.72	2.72	59.42	64.57	1.11
	38	1.00	0.99	0.99	0.98	77.97	3.25	3.25	3.24	3.24	130.41	69.81	1.67
B	30	0.65	0.65	0.64	0.62	6.36	1.42	1.42	1.40	1.40	9.85	55.74	1.55
	40	0.73	0.72	0.70	0.70	24.41	1.60	1.58	1.59	1.55	31.52	54.41	1.29
	45	0.65	0.64	0.61	0.64	46.57	1.61	1.61	1.61	1.59	57.01	59.80	1.22
Avg.		0.96	0.95	0.88	0.88	22.93	2.48	2.46	2.40	2.39	31.60	63.20	1.38

addition, it specifies the vessels' capacity for servicemen $\bar{Q}^1$ and the vessels' maximum allowed spare parts weight $\bar{Q}^2$. The vessels' cost parameter specifies the costs per unit of Euclidean distance traveled, and their speed parameter specifies the units of distance that can be traveled in an hour. For Benchmark A, we let the second vehicle have maximum driving time at least as large as the first vehicle, while for Benchmark B and C we did not make that distinction between the vehicles. Finally, it takes 0.25 time to transfer between a vessel and an offshore location. We incorporated this by increasing the travel time of every arc by 0.25. The instances from each benchmark set are named after the number of services n and time periods T that they consist of, followed by an index, for example, instance An20T5-1 and An20T5-2 are, respectively, the first and second instance with 20 services and 5 time periods in Benchmark Set A. For every combination of the number of services n and the number of periods T, we generated 10 instances and included the first three feasible instances in the benchmarks. The total number of instances is 49 for Benchmark Set A, 18 for Benchmark Set B and 26 for Benchmark Set C.

We tried to solve the instances by directly plugging the compact model formulation (Section 2.1) into CPLEX without including RER inequalities. This led to far worse results than the results presented here; CPLEX was only able to solve the very small instances in reasonable computation times.

2.6.2 Root node relaxations

We study the effect of the RER inequalities of size 1, the subset-row inequalities, and the 2-path inequalities by comparing the optimality gap of the root node relaxations on the instances of Benchmarks A and B. Recall that RER inequalities are bounded in size by the number of vehicles minus one. The impact of RER inequalities of size 1 and 2 is shown by comparing the root node relaxations of Benchmark C for different model formulations. If one of the instances is not solved to optimality by any of the models, the best upper bound is used (see Tables 2.4 - 2.6) to compute the corresponding root node optimality gaps.

The results are given in Tables 2.2 and 2.3. The results are presented according to the number of services the corresponding instances consist of, that is, each row represents averages of the benchmark instances with the corresponding number of services. The results in Table 2.2 below "IPM" are obtained without including RER inequalities of size 1, whereas the results below "AIPM" include RER inequalities of size 1 as model AIPM is based on the column dependent constraints approach. The columns indicated by "none" provide the root node optimality gap in percentages without using additional valid inequalities. The root node optimality gaps are calculated as $(UB - LB_{\text{root}})/LB_{\text{root}} \times 100\%$, where LB_{root} is the lower bound after processing the root node and UB equals the best upper bound (see Tables 4-6). Columns "2p" and "ss" denote the root node optimality gaps with 2-path inequalities and subset-row inequalities, respectively. The root node optimality gap of the full model specification, including both subset-row inequalities and 2-path inequalities, is given in the column "full". Next to that, t_{root} denotes the root node computation time of the full model specification. Differences between IPM and AIPM are given in the columns under "dif.". Here, "full" denotes the percentage decrease of the optimality gap resulting from using AIPM instead of IPM (i.e., the difference between AIPM's and IPM's root node optimality gaps as a percentage of IPM's root optimality gap), and t_{root} indicates the relative speed increase (i.e., t_{root} of IPM divided by t_{root} of AIPM).

Replacing the knapsack type constraints (2.21) with RER inequalities of size 1, that is, using AIPM instead of IPM, results in an average decrease of 63.20% of the root node optimality gap. The time needed for computing the root node relaxations differs significantly among the instances, but on average the root node relaxations of AIPM take 22.93 seconds, whereas the IPM's root node relaxations are obtained in 31.60 seconds on average. The effect of the subset-row inequalities on both AIPM and IPM is noticeable, whereas 2-path inequalities seem less effective.

The results in Table 2.3 show root node optimality gaps for the instances of benchmark C. We compare model formulation IPM (without RER inequalities), IPM

Table 2.3: Optimality gaps (%) of the root node relaxations of benchmark C with different RER inequalities included.

		IPM		IPM + RER1		IPM + RER2		IPM + RER1+2	
n	T	Opt. gap	t_{root}	Opt. gap	t_{root}	Opt. gap	t_{root}	Opt. gap	t_{root}
10	2	1.04	0.08	1.04	0.07	0.61	0.13	0.61	0.13
10	3	0.60	0.06	0.60	0.07	0.57	0.08	0.57	0.08
14	3	1.68	0.21	1.64	0.21	1.40	0.26	1.27	0.27
14	4	1.05	0.23	1.05	0.25	0.94	0.30	0.94	0.29
18	3	1.23	0.63	1.23	0.60	0.63	0.75	0.63	0.75
18	4	1.28	0.45	1.28	0.44	0.84	0.73	0.84	0.69
22	4	1.50	0.95	1.50	0.93	1.22	1.37	1.22	1.35
22	5	1.23	0.96	1.22	0.96	1.04	1.47	1.03	1.55
26	5	2.23	1.25	2.16	1.29	1.53	2.01	1.50	1.54
	avg.	1.31	0.54	1.30	0.54	0.98	0.79	0.96	0.74

with RER inequalities of size 1 (“IPM + RER1”), IPM with RER inequalities of size 2 (“IPM + RER2”), and IPM with RER inequalities of size 1 and 2 (“IPM + RER1+2”). All experiments include both 2-path inequalities and subset-row inequalities. Table 2 presents for each of the four formulations, the root node optimality gap (as calculated in Table 2) and the corresponding calculation time for processing the root node. The formulation “IPM + RER 1+2” provides the best root node relaxations. With an average root node optimality gap of 0.96%, it closes IPM’s root node optimality gap with 26.71%. This is mainly due to the inclusion of RER inequalities of size 2, since they individually close IPM’s optimality gap with 25.19%.

2.6.3 Full model comparison

The solutions to Benchmark sets A, B, and C are given in Tables 2.4 - 2.6, respectively. Each instance is solved with AIPM, which relies on the column dependent constraints approach, and with IPM + RER1 (Benchmark Set A and B) or IPM + RER1+2 (Benchmark Set C) in which the RER inequalities are separated in a traditional way. We included subset-row and 2-path inequalities in all the presented results. The columns “UB”, and “gap” present the best upper bound found and the corresponding optimality gap in percentages, respectively. The optimality gap is calculated as $(UB - LB)/LB \times 100\%$, where LB is the best lower bound. Next, the column “nodes” denotes the number of explored nodes of the branch-and-bound tree, and the column “Sec.” denotes the total runtime. Finally, the percentage difference in runtime is denoted in the column “Δ Sec.”, calculated as [Sec.(AIPM) - Sec.(IPM + RER1)]/[Sec.(IPM + RER1)]× 100 %, and the root node optimality gaps are provide in the last column.

Table 2.4: Solutions to instances benchmark set A with IPM + RER1 and AIPM. RER inequalities of size 1 are included in IPM.

	IPM + RER1				AIPM					
Instance	UB	Gap	Nodes	Sec.	UB	Gap	Nodes	Sec.	Δ Sec.	Gap (root)
An10T3-1	28087.55	0.00	99	2	28087.55	0.00	11	0	-76.70	1.54
An10T3-2	20970.66	0.00	7	0	20970.66	0.00	3	0	-34.88	0.29
An10T3-3	26617.57	0.00	5	0	26617.57	0.00	5	0	-48.98	0.26
An10T4-1	27098.09	0.00	19	0	27098.09	0.00	11	0	-38.89	0.62
An10T4-2	28152.48	0.00	112	2	28152.48	0.00	27	1	-75.00	1.04
An10T4-3	23789.15	0.00	19	0	23789.15	0.00	9	0	-21.95	0.65
An15T4-1	43793.93	0.00	46	3	43793.93	0.00	15	3	-3.23	0.81
An15T4-2	43458.52	0.00	11	1	43458.52	0.00	5	1	32.95	0.43
An15T4-3	48748.17	0.00	182	8	48748.17	0.00	27	4	-47.49	2.23
An15T5-1	43025.34	0.00	41	3	43025.34	0.00	55	4	39.33	0.67
An15T5-2	47227.70	0.00	1	1	47227.70	0.00	1	0	-43.64	0.00
An15T5-3	51964.74	0.00	3	1	51964.74	0.00	3	0	-46.74	0.05
An20T5-1	63872.94	0.00	1824	145	63872.94	0.00	337	126	-13.30	2.07
An20T5-2	38780.62	0.00	1	6	38780.62	0.00	1	11	73.50	0.00
An20T5-3	49958.59	0.00	164	52	49958.59	0.00	51	23	-56.43	1.10
An20T6-1	69517.70	0.00	216	11	69517.70	0.00	68	5	-51.21	1.04
An20T6-2	65297.03	0.00	1	2	65297.03	0.00	1	1	-42.54	0.00
An20T6-3	64001.64	0.00	567	155	64001.64	0.00	201	188	21.19	1.97
An23T6-1	68871.16	0.00	301	92	68871.16	0.00	82	80	-13.60	0.84
An23T6-2	59077.44	0.00	147	43	59077.44	0.00	161	139	221.40	0.79
An23T6-3	62875.43	0.00	27	16	62875.43	0.00	11	15	-5.26	0.38
An23T7-1	68931.47	0.00	164	22	68931.47	0.00	123	32	44.86	0.73
An23T7-2	70931.07	0.00	508	94	70931.07	0.00	145	56	-40.67	0.78
An23T7-3	62269.57	0.00	12	12	62269.57	0.00	9	13	12.04	0.29
An26T7-1	84364.62	0.00	34	24	84364.62	0.00	3	41	73.01	0.12
An26T7-2	72722.87	0.00	7537	2506	72722.87	0.00	1548	1081	-56.85	1.64
An26T7-3	73208.39	0.00	1989	660	73208.39	0.00	864	780	18.04	1.95
An26T8-1	72019.28	0.00	787	128	72019.28	0.00	63	50	-60.88	0.47
An26T8-2	92919.51	0.00	10465	1799	92919.51	0.00	2951	1534	-14.75	1.97
An26T8-3	76139.09	0.00	1602	375	76139.09	0.00	528	401	6.79	1.48
An29T8-1	78543.35	0.00	4247	3157	78543.35	0.00	2167	4010	27.02	1.36
An29T8-2	96981.64	0.00	2488	438	96981.64	0.00	874	492	12.35	0.93
An29T8-3	114259.36	0.00	1591	466	114259.36	0.00	476	469	0.61	1.14
An29T9-1	78185.67	0.00	233	549	78185.67	0.00	117	394	-28.13	1.10
An29T9-2	93382.71	0.00	579	249	93382.71	0.00	141	290	16.28	1.20
An29T9-3	85954.50	0.00	2025	850	85954.50	0.00	953	778	-8.43	1.22
An32T9-1	101388.52	0.00	8447	3352	101388.52	0.00	1542	1078	-67.85	1.37
An32T9-2	111095.36	0.00	1669	2727	111095.36	0.00	1281	4518	65.69	1.38
An32T9-3	100752.91	0.00	798	289	100752.91	0.00	124	186	-35.74	0.95
An32T10-1	84453.77	0.00	3208	6130	84453.77	0.00	987	3554	-42.03	1.43
An32T10-2	123195.44	0.00	3613	3491	123195.44	0.00	1085	1273	-63.55	0.55
An32T10-3	89877.16	0.00	805	2773	89877.16	0.00	443	2607	-6.00	1.05
An35T10-1	119355.72	0.00	1082	1812	119355.72	0.00	442	1144	-36.85	0.60
An35T10-2	115930.11	0.00	1170	1603	115930.11	0.00	501	2193	36.82	0.92
An35T10-3	120197.78	0.00	1435	2757	120197.78	0.00	747	2880	4.45	1.37
An38T10-1	135434.78	0.00	2629	4893	135434.78	0.00	682	2632	-46.20	1.00
An38T10-2	128006.98	0.00	1395	1482	128006.98	0.00	554	2591	74.81	0.97
An38T10-3	135708.68	0.00	1650	9896	135708.68	0.00	760	8794	-11.13	0.96
An41T10-1	196225.73	1.43	6767	10800	196225.73	1.40	2241	10800	0.00	2.97
An41T10-2	133740.76	0.60	4116	10800	134029.73	0.82	967	10800	0.00	1.36
Avg.		0.04	1506.65	1465		0.04	478.51	1296	-5.75	1.00

As can be seen from Table 2.4 and Table 2.5, both methods of handling the RER inequalities provide competitive results. The optimality gaps of both methods are comparable and the computation time of AIPM is slightly lower than the computation time of IPM + RER1, that is, from 1464.80 to 1296.35 seconds for Benchmark A and from 4031.17 to 3132.50 seconds for Benchmark B. This difference is explained by the decrease of the searched branch and bound nodes in the column dependent constraints approach. This is probably caused by a better structured solution space as RER inequalities are generated earlier, which then may increase the efficiency of branching. Furthermore, both methods can solve instances of up to 45 services over a time horizon of 21 days to optimality, showing the strength of the branch-and-price-and-cut algorithm developed. Out of the 117 instances, only five instances remain unsolved within the time limit. Nevertheless, resulting optimality gaps for the 5 unsolved instances are small, that is, on average 0.676 % for AIPM. The root node optimality gaps are on average 1.00% and 0.68% for the instances of Benchmark Sets A and B, respectively. It can be observed that instances that are more difficult to solve suffer indeed from a larger root node optimality gap. However, all resulting gaps are relatively small which can be contributed to the effectiveness of the RER inequalities.

A typical optimal solution of the MSPRP consists of a mixture of routes in which jobs are scheduled simultaneously, sequentially, or a combination of both (see, e.g., Table 7). This is caused by several factors driving the structure of the optimal solutions: 1) minimizing travel costs, 2) minimizing servicemen costs, 3) capacitated vehicles in spare parts and servicemen, and 4) for each job and period exogenous costs are present reflecting the services' relative urgency. This results in a small fraction of the routes being devoted to a single service, i.e., the number of such scheduled services is 8% and 11% for benchmark sets A and B, respectively. The remaining routes' lengths, in the number of visited nodes, equals 4.8 on average over all instances, and routes consisting of 10 nodes are typically part of the optimal solution. This implies that in 40% of the periods a vehicle is not used due to being suboptimal. In addition, we like to stress that during the execution of the branch-and-price-and-cut algorithm routes of large sizes need to be generated, otherwise no optimality guarantee can be provided. These observations and statistics indicate that an enumerative approach, as is taken in Irawan et al. (2018), is indeed not able to solve the MSPRP to optimality.

Table 2.6 presents the result for solving Benchmark Set C. The average number of explored branch and bound nodes is almost four times smaller for AIPM (1004.58 vs. 252.92). Nevertheless, the number of constraints included in the column-dependent constraints approach causes nodes to be processed more slowly, and, consequently, the overall computation time becomes greater in some instances. Finally, all instances could be solved to optimality with both AIPM and IPM + RER1+2.

2.6.4 Ignoring technician costs

Looking into the structure of the optimal solutions, it should be noted that routes of length up to 5 services are observed among the optimal solutions, which is in line with the results presented in Irawan et al. (2017). When technician costs are ignored, the driving factor that disperses services over the time horizon is removed. To assess the performance of the branch-and-price-and-cut algorithm under such circumstances we resolved the instances of Benchmark Set A by removing technician costs. The average optimality gap equals 0.19% on average with an average computation time of 1331 seconds. Comparing withe results in Table 2.4, it is observed that the performance is similar. Hence we can conclude that the branch-and-price-and-cut algorithm developed is able to solve the variant of the MSPRP in which servicemen costs are ignored.

2.6.5 Short service times

To test the performance of the branch-and-price-and-cut algorithm developed, we resolved Benchmark Set B with service times cut in half using AIPM. Although this may not reflect current practices in offshore wind, it might become a reality in a more mature offshore wind industry, in which short inspections and minor repairs might become reality (Willis et al. 2018). We increased the computation time allowed to 24 hours and the maximum memory usage to 64 GB. Average optimality gap after termination equalled 1.10 %, mainly due to instance 2n45T14 (6.8%) for which the memory limit was reached. We clearly observe that only a limited number of periods are being utilized by the vehicles. The average number of routes among the best upper bounds found equals 10.68, with a maximum of 13 routes and a minimum of 8 routes. In Table 7, we provide an example solution (Bn40T21-2) consisting of 11 routes, out of a total possibility of 42 routes (21 periods times 2 vehicles). Looking into the structure of the solution in Table 7, once could clearly observe the mixture of routes in which jobs are scheduled simultaneously (e.g., in Table 7, the route of Vehicle 1 in Period 1) and routes in which jobs are scheduled sequentially (e.g., in Table 7, the route of vehicle 0 in Period 2). This is caused by the several characteristics of the MSPRP influencing the structure of the optimal solution, as is discussed in Section 6.3.

Table 2.5: Solutions to instances benchmark set B with IPM + RER1 and AIPM.

	IPM + RER1				AIPM					
Instance	UB	Gap	Nodes	Sec.	UB	Gap	Nodes	Sec.	Δ Sec.	Gap (root)
Bn30T14-1	255600.40	0.00	3	7	255600.40	0.00	5	6	-13.63	1.00
Bn30T14-2	164668.12	0.00	84	50	164668.12	0.00	44	59	17.04	0.01
Bn30T14-3	229575.13	0.00	546	93	229575.13	0.00	261	80	-13.22	0.22
Bn30T21-1	223300.89	0.00	5613	2255	223300.89	0.00	849	432	-80.82	0.71
Bn30T21-2	198642.60	0.00	85	95	198642.60	0.00	57	44	-53.51	0.96
Bn30T21-3	187745.43	0.00	2282	1741	187745.43	0.00	738	924	-46.93	0.60
Bn40T14-1	274446.27	0.12	13736	10800	274446.27	0.08	6289	10800	0.00	1.22
Bn40T14-2	266032.66	0.00	396	498	266032.66	0.00	179	382	-23.33	1.27
Bn40T14-3	259001.17	0.00	2508	2999	259001.17	0.00	894	2019	-32.67	0.28
Bn40T21-1	293918.58	0.00	1894	1304	293918.58	0.00	1307	1186	-9.00	0.72
Bn40T21-2	316379.51	0.00	82	64	316379.51	0.00	11	31	-51.14	0.96
Bn40T21-3	265363.92	0.17	6281	10800	265363.92	0.00	3106	7727	-28.46	0.33
Bn45T14-1	316761.50	0.73	5003	10801	317716.30	0.80	2922	10800	-0.01	0.68
Bn45T14-2	367451.65	0.00	3004	1629	367451.65	0.00	3475	3733	129.15	1.27
Bn45T14-3	309185.47	0.00	1778	2160	309185.47	0.00	336	961	-55.52	0.57
Bn45T21-1	311858.05	0.00	4220	6466	311858.05	0.00	931	2164	-66.53	0.27
Bn45T21-2	326823.04	0.00	12704	7951	326823.04	0.00	4307	4235	-46.73	0.42
Bn45T21-3	313399.50	0.32	12342	10800	313638.42	0.28	6184	10800	0.00	0.67
Avg.		0.07	4031.17	3917.46		0.06	1771.94	3132.50	-20.85	0.68

Table 2.6: Solutions to instances benchmark set C with IPM and AIPM. RER inequalities of size 1 and 2 included in IPM.

	IPM + RER1+2				AIPM					
Instance	UB	Gap	Nodes	Sec.	UB	Gap	Nodes	Sec.	Δ Sec.	Gap (root)
Cn10T2-1	30463.24	0	23	0	30463.239	0	31	0	81.25	1.60
Cn10T2-2	25253.40	0	3	0	25253.396	0	1	0	-46.15	0.20
Cn10T2-3	29919.01	0	8	0	29919.01	0	1	0	-61.54	0.03
Cn10T3-1	27419.56	0	3	0	27419.559	0	1	0	-41.18	0.00
Cn10T3-2	27250.62	0	1	0	27250.615	0	1	0	-33.33	0.00
Cn10T3-3	29503.38	0	36	0	29503.382	0	20	0	31.25	1.72
Cn14T3-1	38699.91	0	61	1	38699.914	0	41	3	226.83	0.98
Cn14T3-2	41705.84	0	11	0	41705.835	0	3	0	88.89	0.05
Cn14T3-3	41317.37	0	414	2	41317.365	0	143	9	276.76	2.79
Cn14T4-1	40873.76	0	95	1	40873.758	0	26	1	-36.46	0.87
Cn14T4-2	36218.82	0	30	1	36218.821	0	50	5	590.91	1.35
Cn14T4-3	37699.02	0	3	0	37699.022	0	7	0	53.33	0.62
Cn18T3-1	51376.73	0	13	1	51376.725	0	1	2	77.45	0.25
Cn18T3-2	49947.32	0	87	4	49947.321	0	11	5	45.79	1.02
Cn18T4-1	58605.42	0	15	1	58605.42	0	23	1	106.15	0.71
Cn18T4-2	50437.17	0	163	8	50437.17	0	59	8	4.96	1.06
Cn18T4-3	57105.47	0	37	1	57105.47	0	30	2	26.61	0.74
Cn22T4-1	61700.09	0	35	4	61700.09	0	17	5	14.29	0.33
Cn22T4-2	56387.57	0	1489	26	56387.57	0	1964	6028	23428.81	1.91
Cn22T4-3	69951.25	0	6191	370	69951.25	0	879	1277	244.64	1.44
Cn22T5-1	69566.61	0	1233	84	69566.61	0	196	55	-34.43	1.38
Cn22T5-2	68890.12	0	654	15	68890.12	0	219	44	187.17	1.21
Cn22T5-3	75468.34	0	99	3	75468.34	0	101	8	148.84	0.48
Cn26T5-1	81605.927	0	209	11	81605.927	0	35	30	172.94	0.67
Cn26T5-2	80033.842	0	1018	48	80033.842	0	219	282	490.28	1.88
Cn26T5-3	86571.154	0	14188	512	86571.154	0	2497	4860	848.92	1.96
Avg.		0.00	1004.58	42.08		0.00	252.92	485.58	1034.35	0.97

Table 2.7: Solution of instance Bn40T21-2 with small service times.

Vehicle	period	route
0	0	0 25 16 56 65 28 68 81
0	2	0 8 48 35 75 11 51 81
0	4	0 10 50 26 66 27 67 81
0	6	0 12 52 23 63 29 69 81
0	7	0 4 44 5 45 32 72 13 53 81
0	13	0 40 80 34 74 39 79 81
1	0	0 31 71 18 33 73 58 81
1	1	0 24 14 54 64 1 20 60 17 41 2 42 57 81
1	2	0 6 15 55 46 9 49 21 61 81
1	4	0 7 47 38 37 77 78 22 30 62 70 81

2.7 Conclusions

In this paper, we introduced resource exceeding route inequalities, a specialized form of knapsack cover inequalities. These resource-exceeding route inequalities are applicable for any routing problem involving the consumption of a scarce set of resources. Two different approaches for including the resource-exceeding route inequalities were presented. In the first approach, we make use of column-dependent constraints to replace knapsack-type inequalities that model the resource scarcity. We showed that the convex hull of this new formulation is contained in the convex hull of a traditional set-covering formulation for particular cases, and we provided insights in instance characteristics that benefit the most from the new formulation. In order to use the new formulation in a column-generation approach, we formulated an alternative pricing procedure and proved that it provides optimal solutions. A traditional separation procedure for RER inequalities is also presented, in order to assess the computational performance of the column-dependent constraints approach.

The strength of the resource-exceeding route inequalities and the effectiveness of the column-dependent constraints approach has been tested on a new problem in the area of offshore wind maintenance service logistics: the Multi-period Service Planning and Routing Problem. This is the first problem in this area without predefined planning restrictions. A branch-and-price-and-cut algorithm, which is the first sophisticated exact solution approach in the area of offshore wind maintenance service logistics, has been developed to solve the Multi-period Service Planning and Routing Problem. A tailored pulse algorithm with a novel lower bounding procedure was developed to solve the pricing problems.

Computational experiments on a practically inspired set of benchmark instances showed the strength of RER inequalities. Optimality gaps of the root node relaxations on instances with two vehicles were reduced, on average, by 63.20%, a scenario widely encountered throughout the literature. The column-dependent constraints approach appears very effective in searching the branch and bound tree, as compared to a standard separation procedure. Overall, both methods of including RER inequalities are competitive in terms of computational efficiency. Instances of up to 92 nodes and 21 time periods could be solved to optimality in reasonable computation times.

A promising direction for further research is the inclusion of stochastic elements in the Multi-Period Service Planning and Routing Problem. For example, one could model uncertain weather conditions and thereby uncertainty in travel times. Another promising direction is to include what are known as experience-based service times. By doing so, service times become more predictable when servicemen become more experienced in their work.