

**The joint network vehicle routing game
with optional customers**

M. van Zon, G. Desaulniers

G-2020-70

December 2020

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Citation suggérée : M. van Zon, G. Desaulniers (Décembre 2020). The joint network vehicle routing game with optional customers, Rapport technique, Les Cahiers du GERAD G-2020-70, GERAD, HEC Montréal, Canada.

Suggested citation: M. van Zon, G. Desaulniers (December 2020). The joint network vehicle routing game with optional customers, Technical report, Les Cahiers du GERAD G-2020-70, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2020-70>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2020-70>) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Télec. : 514 340-5665
info@gerad.ca
www.gerad.ca

The joint network vehicle routing game with optional customers

Mathijs van Zon ^a

Guy Desaulniers ^b

^a *Econometric Institute, Erasmus University Rotterdam, 3062 PA Rotterdam, The Netherlands*

^b *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

vanzon@ese.eur.nl

guy.desaulniers@gerad.ca

December 2020

Les Cahiers du GERAD

G–2020–70

Copyright © 2020 GERAD, van Zon, Desaulniers

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : We consider logistic collaborations where multiple carriers collaborate by consolidating demands, combining delivery routes, and serving new customers. Logistic collaborations are known to provide numerous benefits such as an increase in profit as well as a reduction in emissions for all parties involved. One of the challenges associated with collaborations is the allocation of the additional profits. To this end, we model the corresponding profit allocation problem as a collaborative game. Here, the profit obtained by any subset of the collaborating carriers depends on the new customers served by the remaining carriers. Specifically, we try to determine an allocation in the core based on both the best-case profit and the worst-case profit that each subset of carriers can attain. To achieve this, we propose a heuristic row generation algorithm. We verify the performance of this algorithm on instances derived from benchmark instances of the capacitated vehicle routing problem. We show that our heuristic algorithm scales well with respect to the number of carriers considered and provides allocations similar to those obtained by enumerating all best-case and worst-case profits of each coalition.

Keywords: Game theory, collaborative transportation, profit allocation, vehicle Routing, row generation, vehicle routing with profits, branch-and-cut-and-price

1 Introduction

Collaborative transportation among carriers has the potential to greatly increase the efficiency of their transportation networks. By collaborating, the carriers can significantly reduce their cost (Guajardo and Rönnqvist, 2015), increase their service levels, and improve their market share (Gansterer and Hartl, 2018). Moreover, such collaborations are known to provide numerous other benefits including reductions in noise pollution, road congestion, and green house gas emissions as shown by Ballot and Fontane (2010) and Pérez-Bernabeu et al. (2014).

In this paper, we study a horizontal collaboration among multiple carriers, each responsible for multiple customers with a given demand. It is assumed that each carrier serves the demand of its customers from a single depot, using a limited fleet of capacitated vehicles. Moreover, we assume that the carriers are rational and aim to maximise their profits. The carriers may collaborate by consolidating demands and combining delivery routes. As a consequence, the carriers are likely to have residual capacity. This residual capacity can be used by the carriers to generate additional profits by collectively serving new customers. We focus on the profit allocation problem that arises when multiple carriers collaborate by consolidating demands, combining delivery routes, and collectively serving new customers.

Profit, or equivalently, cost allocation problems in collaborative transportation are often modelled as cooperative games (Guajardo and Rönnqvist, 2015). A cooperative game is defined as a pair consisting of a set of players, also known as the grand-coalition, and a characteristic function which maps each subset of players, also referred to as a coalition, to a value. We choose to interpret this value as the profit of a coalition, i.e., the value created by the collaboration. We aim to allocate the profit of the grand-coalition to the players such that each player has an incentive to cooperate. To this end, it is common to consider core allocations (Guajardo and Rönnqvist, 2015). A core allocation is an allocation such that the profit allocated to each coalition exceeds the profit which the coalition would be able to obtain without any form of collaboration with the remaining players of the grand-coalition.

Originally, collaborative transportation was studied from the perspective of the customers, rather than the carriers. Göthe-Lundgren et al. (1996), Engevall et al. (1998) and Engevall et al. (2004) consider a setting in which multiple customers are served by a single carrier. More specifically, Engevall et al. (1998) consider the cost allocation problem where all customers are served on a single route, whereas Göthe-Lundgren et al. (1996) and Engevall et al. (2004) consider the cost allocation problem where all customers are served on multiple routes. The aim of the allocation problem is to allocate the cost in a fair manner to each of the customers. To this end, Engevall et al. (1998) model their problem as a travelling salesman game (TSG), which was originally proposed by Fishburn and Pollak (1983). Moreover, Göthe-Lundgren et al. (1996) and Engevall et al. (2004) model their cost allocation problem as a vehicle routing game (VRG), as originally proposed by Göthe-Lundgren et al. (1996).

More recently, collaborative transportation has also been studied from the perspective of carriers. For example, Krajewska et al. (2008) and Dai and Chen (2015) study a profit allocation problem which arises from a collaboration among freight carriers, each with a set of pickup and delivery requests with time windows. In a cooperative setting, the carriers, or players, collaborate by collectively serving their requests. The goal of the corresponding profit allocation problem is to allocate the profit to the carriers, rather than to the customers. Similarly, Zakharov and Shchegryaev (2015) and van Zon et al. (2019) study a collaboration among multiple carriers, each with multiple customers corresponding to delivery requests, which have to be satisfied from a single depot. In contrast to Krajewska et al. (2008) and Dai and Chen (2015), they formulate the problem as a cost allocation problem, with the aim to allocate the cost of the grand-coalition to the carriers.

The previously mentioned studies are mainly concerned with optimising the efficiency of the existing routing networks through collaborative transportation, but do not consider any additional gains which could be obtained as a consequence of the collaboration. For example, in a collaboration among multiple carriers, each carrier has a fleet at its disposal. By consolidating demands and combining delivery routes, the carriers are able to utilise their capacity more efficiently. As a result, less vehicles

may be needed for the transportation and additional customers may be served with their residual capacity for a profit.

The setting of van Zon et al. (2019) is closest to ours. The authors define the cost of a coalition as the optimal objective value to a capacitated vehicle routing problem (CVRP) of the customers belonging to the coalition. They model their cost allocation problem as the joint network vehicle routing game (JNVRG). Moreover, they propose a row generation algorithm to determine a core allocation.

In the previous multiple-player settings, a collaboration between players is established when it becomes profitable for each player to serve customers of other players or to have some of their own customers served by others. In this case, a single customer can easily foster a collaboration between two players as it, generally, does not require a large capacity to be served. On the other hand, it might yield only a small profit increase that might be exceeded by potential overhead collaboration costs.

We believe that a larger profit increase induced by a collaboration can be achieved when the reason for collaborating is to gather sufficient capacity to service an additional large demand. The typical example that we have in mind is the following. To benefit from an economy of scale, a company with multiple branches may require to be supplied by the same carrier. Therefore, to acquire the branches of this company as customers (all together), a carrier must have sufficient capacity or must collaborate with one or several other carriers. This is the type of collaboration that we study in this paper.

We generalize the setting of van Zon et al. (2019) by assuming that there exist multiple clusters of optional customers (each composed, e.g., of branches of a company) which are not assigned to one of the players. A coalition obtains a reward for visiting all customers of a cluster. In this case, we cannot simply define the profit of a coalition, as it is not trivial to determine how the clusters of optional customers are allocated among the coalitions. The allocation of the clusters can be modelled by means of combinatorial auctions (Gansterer and Hartl, 2018). However, by doing so, it is not immediately clear to each of the players what the profit of a coalition is. We choose to consider both a best-case profit as well as a worst-case profit for each coalition. In the best case, we assume that a coalition can freely select the clusters of optional customers it is willing to serve, thereby obtaining the maximum profit. On the other hand, in the worst case, we assume that the remaining players form a coalition and select the clusters of optional customers which benefit them the most. In this case, the original coalition can only select the clusters of optional customers which have not been selected by the remaining players. In this manner, we aim to provide an accurate representation of the profit which the coalition may attain. Our goal is to try to determine an allocation in the core as defined by the best-case profit. However, if this core is empty, we try to determine an allocation in the core as defined by the worst-case profit, or show that the core as defined by the worst-case profit is empty.

Our contributions are as follows. First, we define a new cooperative game which, to the best of our knowledge, has not been studied before. Here, we consider the profits which can be obtained by consolidating demands, combining delivery routes, and collectively serving new customers. For each coalition, we consider both the worst-case and best-case profits that can be obtained, depending on the manner in which the clusters of optional customers are allocated. In the best case, we assume that each coalition can freely select which clusters of optional customers to serve. On the other hand, in the worst case, we assume that the players not included in the coalition get to freely select which clusters of optional customers to serve first. Thereafter, the coalition can only select among the remaining clusters which ones to serve. Secondly, we propose a heuristic row generation algorithm to determine allocations in the cores as defined by the best-case and the worst-case profit, or to show that both cores are empty. The corresponding row generation subproblem is a generalisation of a vehicle routing problem with profits. We propose a branch-and-price-and-cut algorithm for the subproblem. To this end, we adapt several well-known acceleration strategies for the problem.

The remainder of this paper is structured as follows. First, in Section 2, we formally introduce the grouped capacitated profitable tour problem with contract customers (GCPTPCC). Moreover, we introduce the joint network vehicle routing game with optional customers (JNVRGOC), which is a cooperative game based on the GCPTPCC. Here, we also introduce the core as an allocation concept.

In Section 3, we propose a heuristic row generation algorithm in order to determine a profit allocation. We formulate the row generation subproblem as a mixed integer programming problem in Section 3.1 and develop a branch-price-and-cut algorithm for solving the subproblem in Section 3.2. Next, we describe acceleration strategies for the row generation algorithm in Section 3.3. In Section 4, we report numerical results obtained on instances derived from the vehicle routing instances of Augerat (1995). Finally, we present our concluding remarks in Section 5.

2 The joint network vehicle routing game with optional customers

Before presenting the JNVRGOC, we introduce the GCPTPCC which is initially defined for a single carrier. Consider a complete directed graph $G = (V, A)$. The vertex set V is defined as $\{0\} \cup V^c \cup V^o$, where 0 represents the depot of the carrier, V^c the set of contract customers, and V^o the set of optional customers. The set of optional customers V^o is partitioned in m non-empty and disjoint clusters V_l^o with $l \in M = \{1, \dots, m\}$. A reward $p_l^o > 0$ is only obtained if all customers in V_l^o are visited. In contrast to the optional customers, the contract customers in V^c all have to be visited and no reward is given for visiting these customers. With each arc $(v, w) \in A$, a travel cost $c_{vw} \geq 0$ is associated, which we assume to adhere to the triangle inequality. Furthermore, each customer $v \in V^c \cup V^o$ has a demand $d_v > 0$. A limited number of vehicles k with capacity Q is available at the depot to satisfy the demands by visiting customers along routes. A route is defined as a simple cycle, starting and ending at the depot, such that the total demand of the visited customers does not exceed vehicle capacity. Here, we assume that $d_v \leq Q$ for all $v \in V^c \cup V^o$.

The GCPTPCC is the problem of constructing routes in such a way that the profit, i.e., the total rewards minus the total costs, is maximised and every contract customer $v \in V^c$ is visited. We denote the optimal objective value of a GCPTPCC on the graph induced by the vertex sets V^o and V^c as $\text{GCPTPCC}(V^o, V^c)$. To the best of our knowledge, the GCPTPCC has not been studied before. However, van Zon et al. (2019) study the grouped capacitated profitable tour problem (GCPTP), which can be seen as a special case of the GCPTPCC in which no contract customers are considered.

The definition of the GCPTPCC can be generalized to consider multiple collaborating carriers as proposed below. To do so, we need to define one depot per carrier, with its associated vehicle fleet. Nevertheless, in this paper, we assume that there is a single depot for all players which may be seen as a super-depot regrouping the players' individual depots that are assumed to be near each other. The model and algorithm presented below can, however, be easily adapted to the multiple-depot case.

We consider the profit allocation problem that arises when multiple carriers collaboratively serve both contract customers and selected optional customers. Here, we assume that the sets of optional customers as well as the routes used to service them are chosen according to an optimal solution of a GCPTPCC. The overall profit obtained by executing this solution needs to be allocated among the carriers in a fair manner. To this end, we define a cooperative game based on the GCPTPCC, which we call the JNVRGOC. A cooperative game is defined as a pair $\langle N, v \rangle$, where the grand-coalition $N = \{1, \dots, n\}$ denotes the set of players, $v : \mathcal{P}(N) \rightarrow \mathbb{R}$ the profit of a coalition $S \subseteq N$, and $\mathcal{P}(N)$ the power set of N . Here, each player $i \in N$ is assumed to have a finite fleet of k_i vehicles with capacity Q available. Hence, a coalition $S \subseteq N$ has $\sum_{i \in S} k_i$ vehicles available. Moreover, we also assume that each player $i \in N$ has a set of contract customers V_i^c for which the demand has to be fulfilled, with $V_i^c \cap V_j^c = \emptyset$ for $i, j \in N$, $i \neq j$.

The profit of a coalition $S \subseteq N$ depends on how the clusters of optional customers are allocated among the coalition S and the remaining players $N \setminus S$. Here, we consider two specific cases. First, we consider the best-case profit $v^B(S)$ that coalition S can obtain. In the best case, coalition S can freely select clusters of optional customers to serve. We define the best-case profit $v^B(S)$ as follows

$$v^B(S) = \text{GCPTPCC} \left(V^o, \bigcup_{i \in S} V_i^c \right) + \sum_{i \in S} p_i^c, \quad (1)$$

where we add the stand-alone routing costs $p_i^c = -\text{GCPTPCC}(\emptyset, V_i^c)$ of the contract customers belonging to the coalition, in order to ensure that the value created by the coalition is properly represented by $v^B(S)$. Note that $v^B(S)$ is obtained not only by visiting sets of optional clusters but also by visiting more efficiently the contract customers of the players in S . Secondly, we consider the worst-case profit $v^W(S)$ of a coalition $S \subseteq N$, in which we assume that the remaining players $N \setminus S$ form a coalition and can freely select which clusters of optional customers to serve. This may significantly lower the profit that coalition S can obtain. We define the worst-case profit as follows

$$v^W(S) = \text{GCPTPCC} \left(\bigcup_{l \in M'(N \setminus S)} V_l^o, \bigcup_{i \in S} V_i^c \right) + \sum_{i \in S} p_i^c, \quad (2)$$

where $M'(N \setminus S) \subseteq M$ denotes the subset of indices l of the optional customer clusters V_l^o which are not supplied in an optimal solution to $v^B(N \setminus S)$. Notice that there might be several optimal solutions, but we retain only one of them. In this paper, we consider and compare both the best-case profit and the worst-case profit by means of the following profit function

$$v^\alpha(S) = \alpha v^B(S) + (1 - \alpha) v^W(S), \quad (3)$$

where $\alpha \in [0, 1]$ and $S \subseteq N$, such that $v^1(S) = v^B(S)$ and $v^0(S) = v^W(S)$. Moreover, we denote the grand-coalition profit by $v(N)$, as it is independent on the value of α , i.e., $v(N) = v^\alpha(N)$ for any $\alpha \in [0, 1]$. We define the JNVRGOC as a cooperative game $\langle N, v^\alpha \rangle$.

2.1 Profit allocation

The aim of the JNVRGOC is to allocate the profit of the grand-coalition to the players. We denote a profit allocation by $y \in \mathbb{R}^n$, where y_i is the profit allocated to player $i \in N$. Furthermore, we define $y(S) = \sum_{i \in S} y_i$ for $S \subseteq N$. Given $\alpha \in [0, 1]$, a profit allocation $y \in \mathbb{R}^n$ is said to be individually rational if and only if $y_i \geq v^\alpha(\{i\})$ for all $i \in N$. Moreover, y is efficient if and only if $y(N) = v(N)$. Given $\alpha \in [0, 1]$, the set $I(\langle N, v^\alpha \rangle)$ of efficient and individually rational allocations is known as the imputation set. This set is seen as a minimum requirement for cooperation as each player benefits relative to his stand-alone profit. Note that given $\alpha_1 \geq \alpha_2$ it holds that $I(\langle N, v^{\alpha_1} \rangle) \subseteq I(\langle N, v^{\alpha_2} \rangle)$ as $v^{\alpha_1}(\{i\}) \geq v^{\alpha_2}(\{i\})$ for all $i \in N$. Moreover, it may be the case that the imputation set is empty for all $\alpha \in [0, 1]$, as illustrated in the example in Appendix A.

Often, profits (or costs) are allocated by means of core allocations. A core allocation is an allocation which is both efficient and group rational. Here, given $\alpha \in [0, 1]$, an allocation y is said to be group rational if and only if $y(S) \geq v^\alpha(S)$ for all $S \subseteq N$. We denote the core, i.e., the set of all core allocations, by $\text{Core}(\langle N, v^\alpha \rangle)$. Note that $\text{Core}(\langle N, v^{\alpha_1} \rangle) \subseteq \text{Core}(\langle N, v^{\alpha_2} \rangle)$ if and only if $0 \leq \alpha_1 \leq \alpha_2 \leq 1$. Moreover, given $\alpha \in [0, 1]$, it holds that $\text{Core}(\langle N, v^\alpha \rangle) \subseteq I(\langle N, v^\alpha \rangle)$. Next, we provide an example to illustrate the differences between the worst-case core and the best-case core.

Example 2.1 Consider the 3-player instance of the JNVRGOC with $N = \{1, 2, 3\}$ as presented in Figure 1. Let $V_1^c = \{1\}$, $V_2^c = \{2\}$ and $V_3^c = \{3\}$ with $d_1 = d_2 = d_3 = 1$. Moreover, let $k_i = 1$ for each player $i \in N$ and assume that all vehicles have a capacity $Q = 4$. There is a single cluster of optional customers which is defined as $V_1^o = \{4, 5, 6\}$ with $d_4 = d_5 = d_6 = 1$, $p_1^o = 13$. Assuming that $c_{ij} = c_{ji}$ for all $(i, j) \in A$, the relevant arc costs are given in Figure 1.

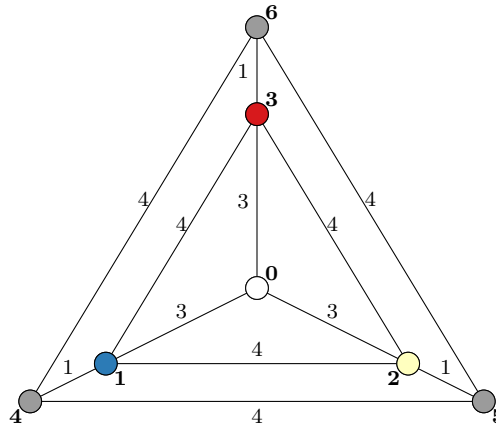


Figure 1: A 3-player instance of the JNVRGOC.

Note that due to the symmetry of the instance, the profits are solely dependent on the size of a coalition. For this reason, we choose a specific coalition of a certain size to determine the profit for all coalitions of this size.

First, we consider the best-case profits. Consider coalition $\{1\}$. In the best case, it is optimal to use route $(0, 1, 4, 5, 6, 0)$ at cost 16. It follows that $v^B(\{1\}) = v^B(\{2\}) = v^B(\{3\}) = 13 - 16 + 6 = 3$. Similarly, we find that for coalitions of size 2, e.g., $\{1, 2\}$, it is optimal to use routes $(0, 1, 4, 6, 0)$ and $(0, 2, 5, 0)$ at costs 12 and 8, respectively. Hence, we find that $v^B(\{1, 2\}) = v^B(\{1, 3\}) = v^B(\{2, 3\}) = 13 - 12 - 8 + 6 + 6 = 5$. For the grand-coalition, it is optimal to use routes $(0, 1, 4, 6, 3, 0)$ and $(0, 2, 5, 0)$ at costs 12 and 8, respectively. Hence, the profit of the grand-coalition is $v^B(N) = 13 - 12 - 8 + 6 + 6 + 6 = 11$.

Secondly, we consider the worst-case profits. For the singleton coalitions, the worst-case profits are trivial ($v^W(\{i\}) = 0, \forall i \in N$) as no optional customers can be visited. On the other hand, it is non-trivial for the coalitions of size 2. For example, consider coalition $\{1, 2\}$. This coalition can be visited using route $(0, 1, 2, 0)$ at cost 10, yielding $v^W(\{1, 2\}) = v^W(\{1, 3\}) = v^W(\{2, 3\}) = 0 - 10 + 6 + 6 = 2$. Hence, despite the fact that no optional customers can be visited, the coalition is still profitable. Finally, recall that the worst-case profit of the grand-coalition is equal to the best-case profit, i.e., $v^W(N) = 11$.

One can observe that, for example, $y^B = (3, 4, 4)$ is a best-case core allocation. On the other hand, $y^W = (9, 1, 1)$ is clearly not a best-case core allocation. However, y^W still belongs to the worst-case core. Both allocations, as well as the worst-case and best-case cores are visualised in Figure 2. At each of the corner points, a single player is allocated a profit of 11 (e.g., player 1 is associated with the lower-left corner), whereas the other players are allocated zero profit. All efficient allocations can be written as convex combinations of these allocations. Hence the largest triangle is the convex hull of these allocations. Each solid line in the figure corresponds to a best-case rationality constraint, whereas each dashed line to a worst-case rationality constraint. Moreover, the arrow indicates the feasible half-space. The best-case core is represented by the dark triangle, whereas the worst-case core is represented by the grey hexagon.

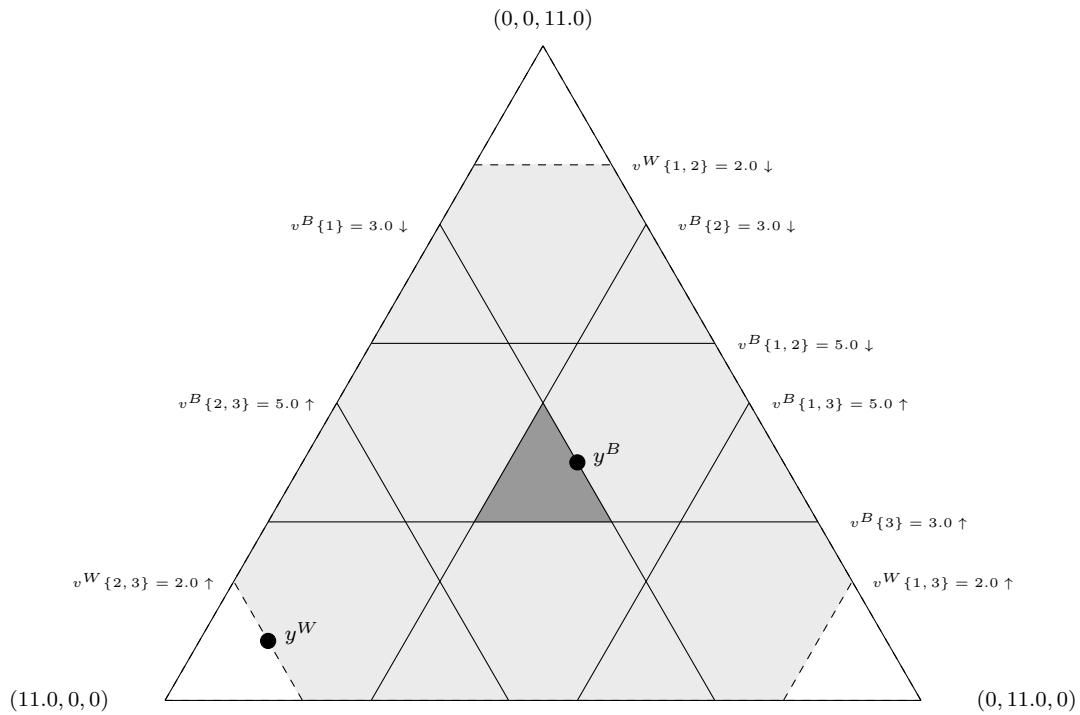


Figure 2: Visualisation of both worst-case and best-case cores.

In the remainder of this paper, we aim to find an allocation for the JNVRGOC in the best-case core. However, if the best-case core is empty, we try to find an allocation in the worst-case core by *minimally* relaxing the best-case rationality constraints. To this end, we consider the following linear programming problem

$$\max \quad \alpha, \tag{4}$$

$$\text{s.t.} \quad y(S) \geq v^\alpha(S) \quad \forall S \subset N, \tag{5}$$

$$y(N) = v(N), \tag{6}$$

$$0 \leq \alpha \leq 1, \tag{7}$$

$$y_i \geq 0 \quad \forall i \in N. \tag{8}$$

To obtain an allocation which is as close as possible to the best-case core, objective function (4) tries to maximise α , which is considered as a decision variable. Constraints (5) enforce the rationality constraints whose right-hand sides are linear functions of α , whereas constraint (6) is the efficiency constraint. Moreover, constraints (7) and (8) define the variable domains.

Note that if there exists an optimal solution α' and y' to (4)–(8) for which $\alpha' = 1$, it holds that y' is in the best-case core. Alternatively, if (4)–(8) has no feasible solution, then the worst-case core is empty.

3 A heuristic row generation algorithm

Two main challenges have to be considered when solving (4)–(8) to optimality. First, there are exponentially many rationality constraints (5) in the number of players. Second, the optimal objective values of two GCPTPCs have to be computed for each coalition. Here, it is important to note that the GCPTPC is a generalisation of the NP-hard CVRP (Lenstra and Kan, 1981), which is well-known to be computationally expensive to solve.

Algorithm 1 Profit allocation heuristic

```

1: Set  $q = 0$ ,  $\Omega_1 = \emptyset$ , and  $\Upsilon_1 = \emptyset$ .
2: Compute  $v(N)$ .
3: repeat
4:    $q = q + 1$ .
5:   Solve (10)–(15) to optimality using row generation and let  $\alpha'_q$  be the optimal value.
6:   if  $\alpha'_q = 1$  then
7:     Set  $\Theta_q = \emptyset$ 
8:   else
9:     Update  $\Upsilon_q$ .
10:    Let  $\Theta_q \subseteq \Upsilon_q$  be the set of coalitions for which constraints (11) are tight.
11:    Let  $\bar{\Theta}_q = \{S \in \mathcal{P}(N) \setminus (\Omega_q \cup \Upsilon_q) \mid N \setminus S \in \Theta_q\}$ .
12:    for  $S \in \Theta_q$  do
13:      Determine  $v^B(S)$ .
14:    end for
15:    for  $S \in \Theta_q$  do
16:      Determine  $v^W(S)$ .
17:    end for
18:    Set  $\Omega_{q+1} = \Omega_q \cup \Theta_q$  and  $\Upsilon_{q+1} = (\Upsilon_q \setminus \Theta_q) \cup \bar{\Theta}_q$ 
19:  end if
20: until  $\Theta_q = \emptyset$ 

```

In similar studies, both challenges are alleviated by applying row generation to the rationality constraints (see Göthe-Lundgren et al., 1996, Engevall et al., 2004, Drechsel and Kimms, 2010, Dai and Chen, 2015, van Zon et al., 2019). Violated rationality constraints are identified by means of a row generation subproblem. Consider an optimal solution α' and y' to (4) and (6)–(8). A violated rationality constraint (5) can be identified by solving the following row generation subproblem

$$\max_{S \subseteq N} \left\{ v^{\alpha'}(S) - y'(S) \right\}. \quad (9)$$

However, if $\alpha' \neq 1$, it is not at all obvious how subproblem (9) can be solved efficiently due to the inclusion of the worst-case profit of coalition S . Alternatively, one could try to apply row generation by using lower bounds on the worst-case profits of the coalitions but it does not seem easy to determine such tight bounds. As a consequence, we were unable to devise an efficient row generation algorithm for solving (4)–(8) to optimality.

For instances with a small number of players, we can simply enumerate both the best-case profits as well as the worst-case profits. However, for instances with a larger number of players, enumeration is no longer tractable. Instead of solving (4)–(8) to optimality, we propose an iterative heuristic algorithm to determine allocations in the best-case core or possibly in the worst-case core, if the best-case core is empty. At each iteration q , this algorithm solves a relaxation of (4)–(8) to determine an allocation. This relaxation is obtained by relaxing the constraints (5) for all coalitions except those in a subset $\Omega_q \subseteq \mathcal{P}(N)$. It writes as follows:

$$\max \quad \alpha_q, \quad (10)$$

$$\text{s.t.} \quad y(S) \geq \alpha_q v^B(S) \quad \forall S \in \mathcal{P}(N) \setminus \Omega_q, \quad (11)$$

$$y(S) \geq \alpha_q v^B(S) + (1 - \alpha_q) v^W(S) \quad \forall S \in \Omega_q, \quad (12)$$

$$y(N) = v(N), \quad (13)$$

$$0 \leq \alpha_q \leq 1, \quad (14)$$

$$y_i \geq 0 \quad \forall i \in N. \quad (15)$$

The objective function (10) still aims at maximizing the value of α which is replaced by α_q at iteration q . Constraints (11) are the relaxed rationality constraints (the term involving $v^W(S)$ is omitted), whereas constraints (12) are the non-relaxed ones. The efficiency constraint is given by (13). Finally, the variable bounds are enforced through (14) and (15).

The proposed profit allocation heuristic is described in Algorithm 1, where Υ_q denotes the set of constraints (11) that are initially considered in model (10)–(15) at iteration q , i.e., those whose

best-case profits have already been computed. We start by setting $q = 0$, $\Omega_1 = \emptyset$, and $\Upsilon_1 = \emptyset$, and we compute $v(N)$ by solving a GCPTPCC for the grand-coalition N . Then, we perform the following steps iteratively, incrementing the iteration counter q at each iteration (Step 4). First, in Step 5, we solve (10)–(15) to optimality using row generation and store the computed optimal value α'_q . To this end, we relax the constraints (11) except those associated with a coalition $S \in \Upsilon_q$ and find any tight or violated best-case rationality constraints by solving a row generation subproblem (see Sections 3.1 and 3.2), which implies computing the best-case profit $v^B(S)$ for each corresponding coalition S . If $\alpha'_q = 1$, it means that an allocation in the best-case core has been found and the algorithm stops after setting $\Theta_q = \emptyset$ in Step 7 to meet the repeat loop exiting criterion in Step 20. Otherwise, after updating set Υ_q in Step 9 by appending the coalitions for which a constraint (11) was generated in Step 5, we identify in Step 10 the best-case rationality constraints (11) that are tight at optimality and define Θ_q as the set of their associated coalitions. For these coalitions, we would like to compute their worst-case profits and convert their corresponding relaxed rationality constraints (11) into non-relaxed constraints (12). To compute these worst-case profits, we need to compute the best-case profits of their complements with respect to N (if not computed yet) and, more specifically, identify the clusters of optional customers that are not covered in the computed optimal solutions. Consequently, in Step 11, we determine the set $\bar{\Theta}_q$ of the complements for which their best-case profits have not been computed yet. Thereafter, for each coalition $S \in \bar{\Theta}_q$, we compute its best-case profit $v^B(S)$ in Step 13 and, for each coalition $S \in \Theta_q$, we find its worst-case profit in Step 16. Finally, we determine Ω_{q+1} and Υ_{q+1} . We stop iterating when $\Theta_q = \emptyset$ and output the last allocation found in Step 5.

Let (α', y') denote the solution computed by this iterative procedure. Moreover, let (α^*, y^*) denote the solution obtained by solving (4)–(8) to optimality. It holds that $\alpha' \geq \alpha^*$. Moreover, $y^* \in \text{Core}(\langle N, v^{\alpha^*} \rangle)$, whereas we can only guarantee $y' \in \text{Core}(\langle N, v^{\alpha'} \rangle)$ if $\alpha' = 1$. Note that this implies that, if $\alpha' < 1$, it is not guaranteed that the allocation is in the worst-case core, i.e., that (α', y') is feasible for (4)–(8). However, each coalition is at least allocated α' times its best-case profit.

3.1 The row generation subproblem

Consider an incumbent solution (α'_q, y') to (10)–(15) at iteration q . We identify tight or violated rationality constraints (11) at iteration q by means of the following row generation subproblem

$$SP(\alpha'_q, y') = \max_{S \in \mathcal{P}(N) \setminus \Omega_q} \{ \alpha'_q v^B(S) - y'(S) \}. \quad (16)$$

If $SP(\alpha'_q, y') \geq 0$, we have identified a tight or violated best-case rationality constraint of the form $y(S) \geq \alpha v^B(S)$. Otherwise, (α'_q, y') is optimal for (10)–(15) and all tight rationality constraints have been generated at iteration q . Note that the profit of a coalition has not yet been determined and is part of solving the subproblem.

As proposed by van Zon et al. (2019), the subproblem can be modelled as a GCPTP where we consider an additional constraint with regards to the number of vehicles used. We formulate the GCPTP as the following mixed integer program. Let R denote the set of feasible routes in $G = (V, A)$. Let c_r be the cost of route r which is given by the sum of the costs of its arcs. Also, let the coefficient a_{rv} be equal to the number of times customer v is included in route r . Hence, $a_{rv} = 0$ if customer v is not included in r and $a_{rv} = 1$ if customer v is included exactly once in r . Next, for each route $r \in R$, define a binary decision variable x_r such that $x_r = 1$ if and only if route r is selected in the solution. Furthermore, for each $l \in M$, we define the binary decision variable z_l such that $z_l = 1$ if and only if all customers of V_l^o are visited exactly once in the solution. For each optional customer $v \in V_l^o$, let $l(v) = l$ and, for each contract customer $v \in V^c$ of player $i \in N$, let $i(v) = i$. Finally, let s_i be a binary variable that indicates whether or not player $i \in N$ belongs to the selected coalition S .

Given this notation, the subproblem can be formulated as follows:

$$SP(\alpha'_q, y') = \max \sum_{l \in M} p_l^o z_l - \sum_{r \in R} c_r x_r + \sum_{i \in N} s_i (p_i^c - \frac{1}{\alpha'_q} y'_i) \quad (17)$$

$$\text{s.t.} \quad \sum_{r \in R} a_{rv} x_r = s_{i(v)} \quad \forall v \in V^c, \quad (18)$$

$$\sum_{r \in R} a_{rv} x_r = z_{l(v)} \quad \forall v \in V^o, \quad (19)$$

$$\sum_{r \in R} x_r \leq \sum_{i \in N} s_i k_i, \quad (20)$$

$$\sum_{i \in S} s_i - \sum_{i \in N \setminus S} s_i \leq |S| - 1 \quad \forall S \in \Omega_q \quad (21)$$

$$x_r \in \{0, 1\} \quad \forall r \in R, \quad (22)$$

$$z_l \in \{0, 1\} \quad \forall l \in M, \quad (23)$$

$$s_i \in \{0, 1\} \quad \forall i \in N. \quad (24)$$

Objective function (17) searches to maximize the amount by which a rationality constraint (11) is violated with respect to the incumbent solution (α'_q, y') . Constraints (18) ensure that all contract customers of the selected players are visited, while (19) impose that all selected optional customers are visited. Constraint (20) guarantees that the number of vehicles available is not exceeded. Moreover, constraints (21) ensure that only coalitions which are not in Ω_q are considered. When seeking all tight constraints (11) in Step 5 of Algorithm 1, the set of constraints (21) also contains constraints associated with previously identified coalitions for which (11) is tight. Finally, the binary requirements on the decision variables are enforced through (22)–(24).

Given a solution (x', z', s') to model (17)–(24), the selected coalition S is given by

$$S = \{i \in N \mid s'_i = 1\}. \quad (25)$$

If (x', z', s') is optimal, its best-case profit is derived from the solution as:

$$v^B(S) = \sum_{l \in M} p_l^o z'_l - \sum_{r \in R} c_r x'_r + \sum_{i \in N} s'_i p_i^c. \quad (26)$$

Note that model (17)–(24) can also be used to compute the values of $v^B(S)$ and $v^W(S)$ whenever necessary. Indeed, $v^B(S)$ with $S \subseteq N$ can be obtained by solving (17)–(24) with $s_i = 1$ for $i \in S$ and $s_i = 0$ for $i \in N \setminus S$. Moreover, given an optimal solution (x', z') to $v^B(N \setminus S)$, $v^W(S)$ can be determined by solving (17)–(24) with $z_l = 0$ for each $l \in M$ such that $z'_l = 1$, $s_i = 1$ for $i \in S$ and $s_i = 0$ for $i \in N \setminus S$.

3.2 A branch-price-and-cut algorithm for the subproblem

We propose a branch-price-and-cut algorithm for solving the row generation subproblem (17)–(24) which potentially involves a large number of route variables. Branch-price-and-cut is a branch-and-bound algorithm where column generation is applied to solve the linear relaxations encountered in the search tree and cuts are added as needed to strengthen them. In this context, the linear relaxation of (17)–(24) is called the master problem. Its solution by column generation proceeds iteratively as follows. At each iteration, a so-called restricted master problem, which is defined as the master problem limited to a subset of its variables, is first solved to yield an optimal pair of primal and dual solutions. Then, to check the optimality of this solution for the complete master problem, a pricing problem that searches for positive reduced cost variables is solved. If such variables are found, they are added to the restricted master problem before starting a new iteration. Otherwise, column generation stops with an optimal solution to the master problem and a valid upper bound for the current branch-and-bound node. A detailed description of the pricing problem is given in Section 3.2.1. To improve the upper bound provided by the master problem solution, the adapted rounded capacity cuts described in Section 3.2.2 are generated. Finally, to derive integer solutions, we apply the branching strategy presented in 3.2.3 if required.

3.2.1 Pricing problem

The column generation pricing problem is used to identify route variables with a positive reduced cost or to show that no such variable exists. Let $\lambda_v \geq 0$, $v \in V^c$, $\mu_v \geq 0$, $v \in V^o$, and σ denote the dual variables associated with constraints (18), (19), and (20), respectively. The reduced cost \tilde{c}_r of a route variable x_r , $r \in R$, can be expressed as

$$\tilde{c}_r = \sum_{(v,w) \in A_r} \tilde{c}_{vw} - \sigma, \quad (27)$$

where $A_r \subset A$ is the subset of arcs included in route r and the modified arc costs \tilde{c}_{vw} for $(v, w) \in A$ are given by

$$\tilde{c}_{vw} = \begin{cases} -c_{vw} + \frac{\lambda_v + \lambda_w}{2} & \text{if } v, w \in V^c \\ -c_{vw} + \frac{\mu_v + \lambda_w}{2} & \text{if } v \in V^o, w \in V^c \\ -c_{vw} + \frac{\lambda_v + \mu_w}{2} & \text{if } v \in V^c, w \in V^o \\ -c_{vw} + \frac{\mu_v + \mu_w}{2} & \text{if } v, w \in V^o \end{cases}. \quad (28)$$

As proposed for many vehicle routing problems (see Costa et al., 2019), we model the pricing problem as an elementary shortest path problem with resource constraints (ESPPRC, see Irnich and Desaulniers, 2005), namely, a load resource to account for vehicle capacity and customer resources for imposing route elementarity. Moreover, we solve the pricing problem using a labelling algorithm in which the labels are stored in buckets corresponding to demand and customer pairs, similar to the algorithm proposed by Martinelli et al. (2014). Here, we enforce route elementarity by means of the unreachable resources introduced by Feillet et al. (2004).

Given that the computational complexity of the ESPPRC is *NP*-hard in the strong sense, we apply several well-known acceleration strategies with regard to the pricing problem. First of all, we relax the master problem by allowing non-elementary routes. More specifically, we consider the *ng*-route relaxation of Baldacci et al. (2011), which has been shown to provide a good trade-off between weaker upper bounds for the master problem and reduced computational effort to solve the pricing problem. Given a predefined neighborhood $U_v \subset V^c \cup V^o$ for each customer $v \in V^c \cup V^o$ (say, its 16 closest customers), a non-elementary route that contains a cycle starting with customer w can be generated if and only if this cycle contains a node v such that $w \notin U_v$. With such a relaxation, the coefficients a_{rv} may take values larger than one.

We also apply decremental state space relaxation as proposed by Righini and Salani (2008). Here, we initially allow non *ng*-routes to be generated by only considering a subset of the unreachable resources. However, if a non *ng*-route is generated, then the subset of unreachable resources is enlarged such that this route can be no longer generated, after which the pricing problem is solved again.

Finally, we first solve the pricing problem in a heuristic manner to find any positive reduced cost variables. Only if no routes have been generated by solving the problem heuristically, we solve the pricing problem to optimality. Moreover, we also terminate the labelling algorithm early if the number of routes with a positive reduced cost which have been identified exceeds a certain threshold. Appendix B describes in detail the column generation process.

3.2.2 Valid inequalities

To limit the number of nodes in the search and usually the overall computational time, it is common to improve the upper bound obtained from the master problem by adding valid inequalities. Before branching, we first consider an adaptation of the well-known rounded capacity cuts (Laporte et al., 1985). Consider a subset of customers $W \subseteq V^c \cup V^o$ and let $k(W) = \lceil \sum_{w \in W} d_w / Q \rceil$ be a lower bound on the number of vehicles required to service the customers in W . The traditional rounded capacity cuts would write as follows:

$$\sum_{(v,w) \in A: v \in W, w \notin W} \left(\sum_{r \in R} b_r^{vw} x_r \right) \geq k(W), \quad (29)$$

where b_r^{vw} indicates how often route $r \in R$ uses arc $(v, w) \in A$. It stipulates that at least $k(W)$ vehicles must traverse an arc entering in W . This inequality would be valid only if all customers in W must be visited. This is not the case for the row generation subproblem. Therefore, we develop a modified version of the rounded capacity cuts that reduce the right-hand side according to the selected players and selected optional customer clusters.

Let $k^o(l) = \lceil \sum_{v \in V_l^o} d_v / Q \rceil$ (resp., $k^c(i) = \lceil \sum_{v \in V_i^c} d_v / Q \rceil$) be a lower bound on the number of vehicles required to service the optional (resp. contract) customers in cluster V_l^o for $l \in M$ (resp., set V_i^c for $i \in N$). For a subset of customers $W \subseteq V^c \cup V^o$, we formulate the corresponding modified rounded capacity cut as follows:

$$\sum_{(v,w) \in A: v \in W, w \notin W} \left(\sum_{r \in R} b_r^{vw} x_r \right) \geq k(W) - \sum_{l \in M(W)} (1 - z_l) k^o(l) - \sum_{i \in N(W)} (1 - s_i) k^c(i), \quad (30)$$

where $M(W)$ denotes the subset of optional customer cluster indices $l \in M$ for which $W \cap V_l^o \neq \emptyset$. Similarly, $N(W)$ is the set of players $i \in N$ for which $W \cap V_i^c \neq \emptyset$. Here, we reduce the total number of vehicles required to serve the customers in W if either an optional customer cluster in $M(W)$ or the contracted customer set of a player in $N(W)$ is not covered exactly once in a solution. In that case, the number of vehicles required for serving W is reduced by a lower bound on the number of vehicles required to serve the customers in the corresponding cluster or set. Note that if $z_l = 1$ for all subsets in $M(W)$ and $s_i = 1$ for all players in $N(W)$, then (30) is equivalent to (29).

Proposition 1 *Let χ be the set of feasible solutions to model (17)–(24). Inequality (30) is valid for $\text{conv}(\chi)$, the convex hull of χ .*

Proof. To show that inequality (30) is valid for $\text{conv}(\chi)$, it is sufficient to show that any solution $S \in \chi$ satisfies (30). Let $\hat{S} = (\hat{x}, \hat{s}, \hat{z}) \in \chi$. Let $W(\hat{S}) \subseteq W$ be the subset of customers in W that are visited in solution \hat{S} and $\bar{W}(\hat{S}) = W \setminus W(\hat{S})$ the subset of those that are not visited. First, let us deduce an upper bound on $k(W)$:

$$\begin{aligned} k(W) &= \left\lceil \frac{\sum_{w \in W} d_w}{Q} \right\rceil \\ &= \left\lceil \frac{\sum_{w \in W(\hat{S})} d_w}{Q} + \sum_{l \in M(W)} \left(\frac{\sum_{w \in \bar{W}(\hat{S}) \cap V_l^o} d_w}{Q} \right) + \sum_{i \in N(W)} \left(\frac{\sum_{w \in \bar{W}(\hat{S}) \cap V_i^c} d_w}{Q} \right) \right\rceil \\ &\leq \left\lceil \frac{\sum_{w \in W(\hat{S})} d_w}{Q} \right\rceil + \sum_{l \in M(W)} \left\lceil \frac{\sum_{w \in \bar{W}(\hat{S}) \cap V_l^o} d_w}{Q} \right\rceil + \sum_{i \in N(W)} \left\lceil \frac{\sum_{w \in \bar{W}(\hat{S}) \cap V_i^c} d_w}{Q} \right\rceil \\ &\leq \left\lceil \frac{\sum_{w \in W(\hat{S})} d_w}{Q} \right\rceil + \sum_{l \in M(W)} \left\lceil \frac{\sum_{v \in V_l^o} d_v}{Q} \right\rceil + \sum_{i \in N(W)} \left\lceil \frac{\sum_{v \in V_i^c} d_v}{Q} \right\rceil \\ &= k(W(\hat{S})) + \sum_{l \in M(W)} (1 - \hat{z}_l) k^o(l) + \sum_{i \in N(W)} (1 - \hat{s}_i) k^c(i). \end{aligned} \quad (31)$$

The first equality comes from the definition of $k(W)$, whereas the second arises from a partition of the customers in W into the customers in $W(\hat{S})$, the uncovered optional customers, and the uncovered contract customers. The first inequality stems from the properties of the ceiling function. The second one ensues from the relations $(\bar{W}(\hat{S}) \cap V_l^o) \subseteq V_l^o$, $l \in M(W)$, and $(\bar{W}(\hat{S}) \cap V_i^c) \subseteq V_i^c$, $i \in N(W)$. Finally, the last equality is obtained from the definitions of $k(W(\hat{S}))$, $k^o(l)$, and $k^c(i)$ combined with the facts that $\hat{z}_l = 0$ for all $l \in M(W)$ and $\hat{s}_i = 0$ for all $i \in N(W)$.

Second, using the upper bound (31), we find the following lower bound on $k(W(\hat{S}))$:

$$k(W(\hat{S})) \geq k(W) - \sum_{l \in M(W)} (1 - \hat{z}_l) k^o(l) - \sum_{i \in N(W)} (1 - \hat{s}_i) k^c(i). \quad (32)$$

Third, because only the customers in $W(\hat{S})$ are visited in \hat{S} , we can replace $k(W)$ by $k(W(\hat{S}))$ in the traditional inequality (29) for W :

$$\sum_{(v,w) \in A: v \in W, w \notin W} \left(\sum_{r \in R} b_r^{vw} x_r \right) \geq k(W(\hat{S})), \quad (33)$$

and use the lower bound (32) to deduce inequality (30). \square

Next, we provide an example to illustrate that the modified capacity inequalities (30) can also cut off part of the feasible region if some of the player variables s_i or the cluster variables z_l are non-integral.

Example 3.1 Consider the 2-player instance of the JNVRGOC with $N = \{1, 2\}$ as presented in Figure 3. Let $V_1^c = \{1\}$ and $V_2^c = \{2\}$ with $d_1 = d_2 = 1$. Moreover, let $k_i = 1$ for each player $i \in N$ and assume that all vehicles have a capacity $Q = 2$. There is a single cluster of optional customers $V_1^o = \{3\}$ with $d_3 = 1$.

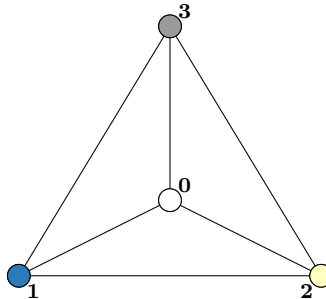


Figure 3: A 2-player instance of the JNVRGOC.

A solution to the linear relaxation of (17)–(24) could have the following non-zero route variables $x_{(0,1,2,0)} = \frac{2}{3}$, $x_{(0,1,3,0)} = \frac{1}{3}$ and $x_{(0,2,3,0)} = \frac{1}{3}$. Let $W = \{1, 2, 3\}$ be such that $s_1 = s_2 = 1$ and $z_1 = \frac{2}{3}$. It follows that the right-hand side of inequality (30) equals $\frac{5}{3}$ while its left-hand side equals $\frac{4}{3}$. Hence, this modified rounded capacity inequality is violated.

To separate the modified rounded capacity cuts, we first run the heuristics provided by the package of Lysgaard et al. (2004) to separate the regular rounded capacity cuts and, then, check if each subset of customers returned yields a violated modified rounded capacity cut. A maximum of 100 cuts is added at a time.

To conclude this section, we would like to mention that we also implemented the limited-memory subset row cuts (lm-SRCs) of size 3, originally proposed by Pecin et al. (2016). In general, the lm-SRCs have shown to provide a good trade-off between an increase in computational time and a decrease in integrality gap (Pecin et al., 2017). However, we observed that they did not improve the computational performance of our algorithm. We believe that this is caused by the relatively small number of customers that can be included in a feasible route for our instances. Therefore, these cuts are not present in the final version of our algorithm.

3.2.3 Branching strategy

To derive integer solutions, we use the following hierarchical branching strategy. When an optimal solution to the master problem is fractional and branching is required, this strategy selects the first branching entity taking a fractional value according to the following order:

1. The sum of the player variables $\sum_{i \in N} s_i$;
2. A player variable s_i ;
3. The sum of the cluster variables $\sum_{l \in M} z_l$;
4. A cluster variable z_l ;
5. The total number of vehicles used $\sum_{r \in R} x_r$;
6. An arc-flow variable $\sum_{r \in R} b_r^{vw} x_r$.

When multiple entities are available, the one taking the closest fractional value to 0.5 is chosen. Moreover, when choosing a node to branch on, we choose the node with the largest upper bound.

3.3 Acceleration strategies

In this section, we describe two acceleration strategies that can be applied to the basic row generation algorithm described in Section 3. First, throughout the algorithm, we store all generated columns in a pool which is shared among all branch-price-and-cut procedures. Hence, this pool is not only used when solving row generation subproblems, but also when determining best-case profits and worst-case profits. In this manner, we can generally provide a good initial set of columns for any call to a branch-price-and-cut algorithm.

Secondly, for the row generation subproblem, we terminate the branch-price-and-cut procedure as soon as a coalition corresponding to a violated or tight rationality constraint is identified, when an integral solution with a non-negative value is found. Here, it should be noted that the profit which has been determined for this coalition is not necessarily optimal. Hence, we still need to determine the best-case profit of the identified coalition, after which we add the corresponding best-case rationality constraint.

4 Computational results

In this section, we first describe the instances used for our computational experiments and present our experimental plan. Then, we report the computational results obtained and discuss the quality of the computed solutions.

4.1 Instances and experimental setup

We constructed instances of the JNVRGOC based on the CVRP instances of Augerat (1995) as follows. Specifically, we considered the instances of the A-set which involve between 31 and 79 customers in total. For each of these instances, we defined nine JNVRGOC instances by considering 3, 5 and 7 players as well as 3, 5 and 7 clusters of optional customers. Recall that the number of customers is given by $|V|-1$. We assigned the first $n \lfloor \frac{0.7(|V|-1)}{n} \rfloor$ customers to the players, such that $i(v) = (v \bmod n) + 1$ with $v \in \{1, \dots, n \lfloor \frac{0.7(|V|-1)}{n} \rfloor\}$. The remaining customers $\{n \lfloor \frac{0.7(|V|-1)}{n} \rfloor + 1, \dots, |V|\}$ were divided among the clusters of optional customers such that $l(v) = \left[\left(v - \left(n \lfloor \frac{0.7(|V|-1)}{n} \rfloor + 1 \right) \right) \bmod m \right] + 1$ with $v \in \{n \lfloor \frac{0.7(|V|-1)}{n} \rfloor + 1, \dots, |V|\}$.

Consider an instance of the JNVRGOC with players $\{1, \dots, n\}$, contract customers V^c and optional customers V^o . For each instance, we assign each player $i \in N$ precisely enough vehicles to cover its

own customers: $k_i = \left\lceil \frac{\sum_{v \in V_i^c} d_v}{Q} \right\rceil$. Moreover, given a cluster of optional customers V_l^o with $l \in M$, we define the reward p_l^o as $-0.8 \cdot \text{GCPTPCC}(\emptyset, V_l^o)$, where $-\text{GCPTPCC}(\emptyset, V_l^o)$ denotes the optimal value of a CVRP defined over the customers in V_l^o .

We solved these instances using four algorithms. The first algorithm, denoted by RG, is the row generation algorithm described in Section 3.2. To determine the performance of the modified capacity cuts (30), we also consider an algorithm, denoted by RG NO MCC, which is the row generation algorithm RG but excluding the modified rounded capacity cuts. Moreover, we also execute two enumeration algorithms ENUM RG and ENUM ALL. The former enumerates a priori the best-case profit for each coalition, and only determine the worst-case profit for coalitions associated with tight rationality constraints at each iteration of Algorithm 1. In the latter, we enumerate the best-case profit and the worst-case profit for each coalition, which allows us to solve (4)–(8) to optimality. Algorithms ENUM RG and ENUM ALL are used as benchmarks with respect to computation times and optimal solution values, respectively.

All algorithms were implemented in C++, using IBM ILOG Optimiser 12.8.0 to solve any linear programming problems. The computations were conducted on a single thread of an Intel Xeon Gold 6130 @ 2.1 GHz with a turbo of 3.7 GHz and 16 GB of RAM. Unless otherwise specified, a time limit of 7,200 seconds (2 hours) is enforced for solving each instance.

4.2 Computational performance

To compare the performance of the four tested algorithms, we only consider 10 of the smallest Augerat instances, with only 3 and 7 players, and 3 and 7 clusters of optional customers. Specifically, we consider Augerat instances A-n32-k05, A-n33-k06, A-n34-k05, A-n36-k05, A-n37-k06, A-n38-k05, A-n39-k05, A-n39-k06, A-n44-k06 and A-n45-k06, leading to a total of 40 JNVRGOC instances.

Summarised results obtained for these small instances are shown in Table 1. They are grouped by the number of players n and the number of optional customer clusters m . For each algorithm, two columns are presented to report the number of instances solved to optimality within the time limit (OPT), and the average time taken to solve these instances (T).

Table 1: Comparative results on small JNVRGOC instances.

n	m	RG NO MCC		RG		ENUM RG		ENUM ALL	
		OPT	$T(s)$	OPT	$T(s)$	OPT	$T(s)$	OPT	$T(s)$
3	3	8	2783	10	403	10	279	10	299
3	7	8	1757	10	388	10	249	10	278
7	3	2	2007	8	3124	1	3457	1	6339
7	7	2	1648	9	4040	1	3297	1	6054

For each (n, m) combination, we were able to solve significantly more instances with algorithm RG than with algorithm RG NO MCC. For example, we solved 8 of the 10 instances with 7 players and 3 clusters of optional customers with algorithm RG, compared to only 2 instances with algorithm RG NO MCC. This clearly demonstrates the effectiveness of the modified capacity cuts. Moreover, the effectiveness of the row generation algorithm is illustrated by comparing algorithms RG NO MCC and ENUM RG. Here, we can see that even without the capacity cuts, algorithm RG NO MCC is able to solve more instances than ENUM RG for the 7 player instances. For these instances, the difference is much more striking when comparing algorithms RG and ENUM RG

Furthermore, it can be seen that algorithm RG does not outperform the enumeration algorithms ENUM RG and ENUM ALL for instances with 3 players. For these instances, only 6 coalitions have to be considered, which causes most, if not all, coalitions to be generated by the row generation algorithm with the additional overhead of solving the row generation subproblem multiple times. However, for the

7-player instances, we are able to solve significantly more instances with algorithm RG in significantly less time when compared to ENUM RG and ENUM ALL. Here, it should be noted that ENUM ALL is an exact algorithm whereas algorithms RG and RG ENUM may yield a solution that is not feasible to (4)–(8).

Next, we discuss the performance of algorithm RG in more detail. A summary with respect to the performance of this algorithm for each combination (n, m) is provided in Table 2. We consider the following summary statistics: the number of instances solved (OPT), the average run time in seconds (T), and the average number of rationality constraints of the form (11) that have been generated (I). Moreover, we also consider the proportions of the time spent solving the row generation subproblems (T_i) and all other routing problems (T_g), as well as the average number of best-case (BP) and worst-case (WP) problems solved. Finally, we also list the average number of branch-and-bound nodes explored (BBN) and the modified capacity cuts generated (MCC). It should be noted that the averages are computed only over the instances which were solved before reaching the time limit. The complete computational results for each individual instance can be found in Appendix C.

Table 2: Summary results of the algorithm rg on the JNVRGOC instances.

n	m	OPT	$T(S)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC
3	3	21	1014	2.1	59	41	5.1	2.0	666	485
3	5	19	1274	2.6	53	47	5.5	2.3	843	511
3	7	19	1277	2.7	57	43	5.5	2.6	642	634
5	3	12	1569	7.7	55	44	12.8	4.4	1395	689
5	5	13	1910	8.2	56	44	13.3	5.0	1577	838
5	7	12	2193	11.4	57	43	15.7	4.9	1489	870
7	3	9	3098	17.3	64	36	24.8	7.0	2720	1280
7	5	8	3363	20.1	59	41	27.4	7.1	2628	1117
7	7	9	4040	21.0	56	44	27.1	7.1	2694	1151

On average 42%, 30% and 15% of all possible coalitions were generated by algorithm RG on the 3, 5 and 7-player instances that have been solved, respectively. Here, it should be noted that the maximum number of coalitions which could have been generated is equal to 6, 30 and 126 for the 3, 5 and 7-player instances, respectively. Clearly, the percentage of coalitions generated decreases as the number of players increases, which illustrates the effectiveness of a row generation procedure for instances with a relatively high number of players. Moreover, the average size of coalitions generated is 1.7, 3.0 and 3.4 for the 3, 5 and 7-player instances, respectively.

The time spent on the heuristic is almost split equally between solving the row generation subproblems and determining best-case and worst-case profits. However, we can observe that less subproblems are solved than the number of profits computed, which shows that computing profits is on average less time-consuming. This is not surprising given that, to do so, some player and cluster variables are fixed. Moreover, it is interesting to note that, for 3-player instances, the best-case profits have been determined for almost every coalition. This is clearly not the case for the 5 and 7-player instances. Finally, we can observe that the total numbers of branch-and-bound nodes explored and cuts generated increase with the number of players but not with the number of optional customer clusters.

To conclude this section, let us remark that we also considered instances in which each player was assigned an additional vehicle compared to the minimum number of vehicles required. However, these instances did not provide any additional insights and, therefore, their results have not been included in this paper.

4.3 Solution quality

In this section, we assess the quality of the solutions produced by algorithm RG. To do so, we compare them with the optimal solutions obtained for model (4)–(8), namely, those computed by algorithm ENUM RG. To find more optimal solutions with this algorithm and, thus, make a comparison on a

larger number of instances, we have increased its time limit to 36,000 seconds. The time limit imposed for algorithm RG has, however, remained equal to 7,200 seconds.

The results are summarised in Table 3. For each combination of n and m , 26 instances were considered. For each algorithm, we report the number of instances solved to optimality within their respective time limit (OPT) and the average objective value of the computed solution, denoted α' for algorithm RG and α for algorithm ENUM ALL. Moreover, when algorithm ENUM ALL was able to compute the best-case and worst-case profits of all coalitions, it becomes possible to determine if the allocation obtained by algorithm RG is feasible for model (4)–(8) and, if so, what is the objective value of this allocation according to this model. The columns SH and INF indicate the number of instances for which both algorithms terminated before their respective time limit and, among these instances, the number of allocations computed by algorithm RG that are infeasible for model (4)–(8), respectively. Indeed, such an allocation y can be infeasible if there exists no α value such that all constraints (5) hold, i.e., if there exists a coalition S for which $y(S) < v^W(S)$. Finally, for both algorithms, column α_{sh} reports the average value of the computed coalitions according to model (4)–(8). Because algorithm ENUM ALL computes an optimal solution, the value of α_{sh} obtained by algorithm RG is always less than or equal to that derived by algorithm ENUM ALL. The gap between these two values indicates the quality of the solution computed by algorithm RG. Again, the complete results for each instance are provided in Appendix C.

Table 3: Summarised results on all JNVRGOC instances.

n	m	RG					ENUM ALL		
		OPT	α'	SH	INF	α_{sh}	OPT	α	α_{sh}
3	3	21	1.00	21	0	1.00	24	1.00	1.00
3	5	19	0.96	19	1	0.96	21	0.96	0.97
3	7	19	0.91	19	1	0.90	23	0.92	0.91
5	3	12	0.91	12	3	0.95	13	0.89	0.96
5	5	13	0.81	13	5	0.79	14	0.73	0.86
5	7	12	0.77	11	4	0.64	12	0.71	0.80
7	3	9	0.71	3	2	0.56	3	0.61	0.63
7	5	8	0.65	4	1	0.38	4	0.59	0.56
7	7	9	0.63	4	1	0.46	4	0.56	0.53

From these results, we make the following observations. First, if we consider the objective value α' , we see that players are on average allocated at least 96%, 83% and 66% of their best-case profit for the 3, 5 and 7-player instances, respectively. Here, it is interesting to note that players are on average allocated less profit relative to their best-case profit if the size of the grand-coalition increases. A similar pattern occurs with respect to the optimal value α , which is on average 96%, 78% and 58% for the 3, 5 and 7-player instances, respectively. We believe that the decrease in relative allocated profit is partially accounted for by the structure of the instances. For example, consider a given instance with a fixed number of optional customer clusters. Recall that each player has precisely enough vehicles to cover its contract customers. Hence, the overall number of vehicles is likely to be larger for a larger number of players as the customers are served less efficiently. The larger number of vehicles may cause smaller coalitions to already be able to serve most clusters of optional customers, thereby reducing the value created by having many players collaborating. As a consequence, each coalition is allocated less when compared to their best-case profit.

Moreover, we observe that the worst-case profit core is non-empty for all instances solved by the algorithm ENUM ALL. This suggests that the allocation is suitable for use in practice as there is always an incentive to cooperate. Here, it should be noted that the allocation computed by algorithm RG may not be in the worst-case profit core, even if it is non-empty. Specifically, in 3%, 33% and 36% of the cases, this allocation is not in the worst-case profit core for the 3, 5 and 7-player instance, respectively, as indicated by the values reported in the columns SH and INF . However, for the allocations that are in the worst-case core, the objective value is comparable to that of the optimal allocation com-

puted by algorithm ENUM ALL (see both columns α_{sh}). The average deviations between these values are 1%, 7% and 13% on average for the 3, 5 and 7-player instances, respectively.

For each computed allocation, the carriers are given an incentive to collaborate as they are allocated at least as much profit when compared to their worst-case profit. However, in theory, the collaboration between multiple carriers may be hindered if an allocation is proposed which is not in the worst-case core. If this is the case, a subset of carriers may want to leave the collaboration. Nonetheless, in practice, it is very difficult for a subset of carriers to verify whether an allocation is in the worst-case core. Moreover, if a subset of carriers identifies that a given allocation is not in the worst-case core, the corresponding violated rationality constraints can be incorporated after which a new allocation can be easily determined.

5 Concluding remarks

We proposed a heuristic row generation algorithm to determine core allocations for the JNVRGOC, based on both the worst-case profit and the best-case profit. Our computational experiments have shown that an enumeration algorithm which requires computing the worst-case and best-case profits for all possible coalitions becomes rapidly inefficient as the number of players increases. At the opposite, the performance of our row generation approach is less impacted by such an increase. Moreover, our results suggest that the modified capacity cuts do significantly reduce the computation time for the row generation algorithm. As for the quality of the solutions produced by the proposed heuristic, we find that the average deviation of their values with respect to those of the optimal allocations is relatively small. On the other hand, our approach might sometimes fail to find an allocation in the worst-case core when it is not empty.

In order to improve the performance of our algorithm it is crucial to be able to estimate good lower bounds for the worst-case profits. It would require further research to determine such lower bounds.

Appendix

A Emptiness of the imputation set

Consider the 3-player instance of the JNVRGOC with $N = \{1, 2, 3\}$ as presented in Figure 4. Let $V_1^c = \{1\}$, $V_2^c = \{2\}$ and $V_3^c = \{3\}$ with $d_1 = d_2 = d_3 = 1$. Moreover, let $k_i = 1$ for each player and assume all vehicles to have a capacity of 2. The set of optional customers is defined such that $V_1^o = \{4\}$ and $V_2^o = \{5\}$ with $d_4 = 1, d_5 = 2, p_1^o = 5$ and $p_2^o = 8$, respectively. Assuming that $c_{ij} = c_{ji}$ for all arcs $(i, j) \in A$, the arc costs are given in Figure 4 as edge costs (those not specified are equal to 2).

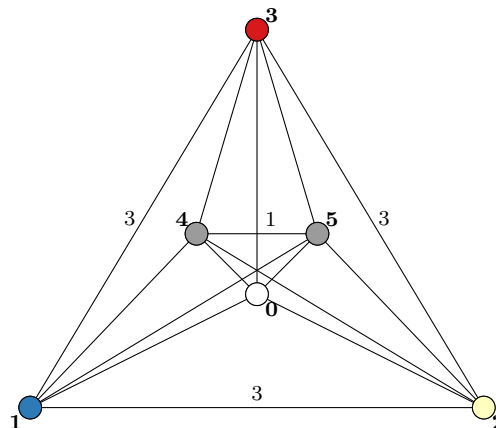


Figure 4: A 3-player instance of the JNVRGOC.

We will show that the imputation set is empty with respect to the worst-case profit, i.e., $\alpha = 0$. In order to determine the worst-case profit $v^W(\{1\}) = v^W(\{2\}) = v^W(\{3\})$, we first have to determine the best-case profit of the coalitions of size two, i.e., we have to determine $v^B(\{1, 2\}) = v^B(\{1, 3\}) = v^B(\{2, 3\})$. Given a two-player coalition, it holds that it is optimal to visit optional customer 5. For example, consider the coalition $\{1, 2\}$. Using routes $(0, 1, 2, 0)$ and $(0, 5, 0)$, a reward of 8 is collected, while the routes cost 7 and 4, respectively. Moreover, note that $p_1^c = p_2^c = p_3^c = 4$. Hence, $v^B(\{1, 2\}) = v^B(\{1, 3\}) = v^B(\{2, 3\}) = 8 - 7 - 4 + 4 + 4 = 5$. It follows that the singleton coalitions $\{1\}$, $\{2\}$ and $\{3\}$ are only allowed to visit optional customer 4. In the worst-case solution, a singleton coalition visits its contract customers as well as optional customer 4 on the same trip for a reward of 5. Hence, $v^W(\{1\}) = v^W(\{2\}) = v^W(\{3\}) = 5 - 6 + 4 = 3$. The optimal profit of the grand-coalition is given by visiting every customer using the routes $(0, 1, 4, 0)$, $(0, 2, 3, 0)$ and $(0, 5, 0)$ at a routing cost of $6 + 7 + 4$. This results in a profit $v^B(N) = v^W(N)$ of $5 + 8 - 6 - 7 - 4 + 4 + 4 + 4 = 8$. Hence, it follows that $v^W(\{1\}) + v^W(\{2\}) + v^W(\{3\}) = 9 > 8 = v^W(N)$, which implies that the imputation set $I(\langle N, v^0 \rangle)$ is empty, and thus $I(\langle N, v^\alpha \rangle) = \emptyset$ for all $\alpha \in [0, 1]$.

B Details on the column generation algorithm

The column generation algorithm proceeds as follows. At each iteration, the restricted master problem is solved. If the number of columns it contains exceeds 5000, then all positive reduced cost columns are removed and stored in a pool of columns. Then we seek negative reduced cost columns using different procedures. As soon as one procedure finds negative reduced cost columns, these columns are added to the restricted master problem which is then re-optimized to start a new iteration. The procedures, which are called in the following order, are as follows:

1. Check if any of the columns in the pool of previously removed columns have a negative reduced cost;
2. Apply heuristic labelling algorithm HLA1;
3. Apply heuristic labelling algorithm HLA2;
4. Apply heuristic labelling algorithm HLA3;
5. Apply heuristic labelling algorithm HLA4;
6. Apply heuristic labelling algorithm HLA5;
7. Apply heuristic labelling algorithm HLA6;
8. Apply the exact labelling algorithm.

All labelling algorithms solve the ng -route relaxed pricing problem, where the neighbourhoods consist of the 16 closest customers. In heuristics HLA1 to HLA4, we do not apply decremental state-space relaxation because dominance is performed only on the reduced cost of a route and a network of reduced size is considered. In HLA1 to HLA4, we restrict the arcs to the arcs between each customer and its 2, 4, 8 and 12 closest neighbours, respectively. In HLA5, HLA6 and the exact labelling algorithm, decremental state-space relaxation is applied as full dominance is considered. However, in HLA5 and HLA6, a reduced network is considered, namely, we restrict the arcs to the arcs between each customer and its 2 and 4 closest neighbours, respectively.

C Detailed results for the algorithms *RG* and *ENUM ALL*

In Tables C.1–C.9, we present detailed results for each instance solved with algorithms *RG* and *ENUM ALL*. For each instance, we report the following statistics for algorithm *RG*:

- α' : The objective value returned by algorithm *RG*;
- α : The objective value of the computed allocation with respect to (4)–(8);
- T : The overall computation time in seconds;
- I : The number of rationality constraints of the form (11) generated;
- T_i : The percentage of time taken to solve the row generation subproblem;

T_g : The percentage of time taken to solve all routing problems excluding the row generation sub-problem;

BP : The number of best-case profits computed, including that for the grand-coalition;

WP : The number of worst-case profits computed;

BBN : The overall number of nodes explored in all branch-price-and-cut procedures;

MCC : The overall number of modified capacity cuts generated.

For algorithm ENUM ALL, we only report the α value and the time T in seconds.

Table C.1: Results on JNVRGOC instances with 3 players and 3 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	1.00	1.00	72	3	85	14	6	2	101	112	1.00	26
A-n33-k06	1.00	1.00	32	2	43	57	5	2	222	166	1.00	23
A-n34-k05	1.00	1.00	37	2	52	47	5	2	83	95	1.00	50
A-n36-k05	1.00	1.00	295	2	83	17	5	2	199	106	1.00	92
A-n37-k05	1.00	1.00	108	2	74	26	5	2	116	165	1.00	46
A-n37-k06	1.00	1.00	1207	2	26	74	5	2	2336	749	1.00	1083
A-n38-k05	1.00	1.00	54	2	75	25	5	2	51	58	1.00	20
A-n39-k05	1.00	1.00	1351	2	28	71	5	2	1240	678	1.00	1158
A-n39-k06	1.00	1.00	116	3	71	29	6	2	121	177	1.00	38
A-n44-k06	1.00	1.00	397	2	78	22	5	2	400	465	1.00	239
A-n45-k06	1.00	1.00	464	2	61	39	5	2	503	408	1.00	263
A-n45-k07	1.00	1.00	2436	2	20	80	5	2	1910	1072	1.00	3294
A-n46-k07	1.00	1.00	877	2	51	49	5	2	391	965	1.00	795
A-n48-k07	1.00	1.00	271	2	80	20	5	2	143	269	1.00	105
A-n53-k07	—	—	—	0	0	100	1	0	3217	609	—	—
A-n54-k07	1.00	1.00	1377	2	73	27	5	2	134	785	1.00	326
A-n55-k09	1.00	1.00	199	2	78	21	5	2	98	457	1.00	64
A-n60-k09	1.00	1.00	1110	2	72	28	5	2	242	591	1.00	425
A-n61-k09	1.00	1.00	1926	2	69	31	5	2	1660	996	1.00	1381
A-n62-k08	—	—	—	2	33	67	5	1	1110	1537	1.00	4528
A-n63-k09	—	—	—	1	5	95	2	0	2349	1340	1.00	8006
A-n63-k10	1.00	1.00	1484	2	19	81	5	2	697	1029	1.00	1349
A-n64-k09	1.00	1.00	6758	2	8	92	5	2	2993	498	1.00	10347
A-n65-k09	1.00	1.00	726	2	88	12	5	2	354	340	1.00	180
A-n69-k09	—	—	—	2	12	88	3	0	2956	441	1.00	6987
A-n80-k10	—	—	—	0	0	100	1	0	1203	1445	—	—

Table C.2: Results on JNVRGOC instances with 3 players and 5 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	1.00	1.00	53	2	75	25	5	2	89	111	1.00	24
A-n33-k06	0.81	0.81	84	3	31	67	7	3	470	197	0.81	53
A-n34-k05	1.00	1.00	33	2	58	42	5	2	64	190	1.00	24
A-n36-k05	1.00	1.00	91	2	84	16	5	2	72	82	1.00	29
A-n37-k05	0.98	0.98	150	3	79	21	5	3	121	100	0.98	64
A-n37-k06	0.84	0.72	1807	5	19	81	6	3	3358	835	0.83	1850
A-n38-k05	1.00	1.00	146	2	37	63	5	2	187	110	1.00	107
A-n39-k05	0.86	x	620	4	56	43	6	3	528	600	0.86	456
A-n39-k06	0.91	0.91	368	3	52	46	7	3	276	139	0.91	191
A-n44-k06	1.00	1.00	256	2	85	15	5	2	126	499	1.00	58
A-n45-k06	1.00	1.00	213	2	36	63	5	2	239	407	1.00	193
A-n45-k07	1.00	1.00	886	2	36	64	5	2	1094	911	1.00	733
A-n46-k07	0.87	0.87	500	3	63	35	7	3	318	398	0.87	285
A-n48-k07	1.00	1.00	3567	2	17	83	5	2	3060	1044	1.00	3960
A-n53-k07	—	—	—	0	0	100	1	0	4553	710	—	—
A-n54-k07	—	—	—	3	24	76	6	1	2207	1162	1.00	15106
A-n55-k09	1.00	1.00	257	2	68	31	5	2	198	325	1.00	131
A-n60-k09	1.00	1.00	6215	2	52	48	5	2	1201	1169	1.00	2992
A-n61-k09	1.00	1.00	434	2	69	30	5	2	165	580	1.00	1488
A-n62-k08	—	—	—	0	11	89	2	0	1691	972	—	—
A-n63-k09	—	—	—	0	0	100	1	0	2249	1754	—	—
A-n63-k10	1.00	1.00	6770	3	19	81	5	2	4066	1305	1.00	11710
A-n64-k09	—	—	—	2	5	93	3	0	2677	593	1.00	24825
A-n65-k09	—	—	—	0	0	100	1	0	3017	588	—	—
A-n69-k09	1.00	1.00	1749	3	69	31	6	2	383	711	1.00	885
A-n80-k10	—	—	—	0	94	6	1	0	868	1107	—	—

Table C.3: Results on JNVRGOC instances with 3 players and 7 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.92	0.88	69	3	63	36	5	3	142	157	0.92	35
A-n33-k06	0.70	0.70	21	3	57	39	7	3	94	151	0.70	10
A-n34-k05	0.96	0.96	36	3	61	38	5	3	82	85	0.96	25
A-n36-k05	1.00	1.00	147	2	81	18	5	2	122	190	1.00	73
A-n37-k05	0.94	x	199	3	86	14	5	3	132	145	0.94	60
A-n37-k06	0.81	0.81	1148	3	23	75	7	3	1864	1516	0.81	1055
A-n38-k05	0.92	0.92	181	3	45	54	5	3	169	84	0.92	133
A-n39-k05	0.72	0.72	1460	4	52	48	6	3	858	780	0.72	1080
A-n39-k06	1.00	1.00	138	2	80	20	5	2	157	184	1.00	111
A-n44-k06	0.81	0.81	428	3	61	35	7	3	418	555	0.81	151
A-n45-k06	0.83	0.69	249	4	69	31	6	3	153	477	0.79	105
A-n45-k07	—	—	—	0	0	100	1	0	5161	1146	—	—
A-n46-k07	0.81	0.70	618	4	79	21	6	3	254	725	0.81	285
A-n48-k07	0.96	0.96	848	3	85	15	5	3	385	608	0.96	1214
A-n53-k07	1.00	1.00	2404	2	39	61	5	2	949	1132	1.00	4604
A-n54-k07	—	—	—	0	13	87	2	0	3094	1392	0.93	17040
A-n55-k09	1.00	1.00	1267	2	33	67	5	2	1295	584	1.00	1064
A-n60-k09	1.00	1.00	1343	2	72	28	5	2	417	715	1.00	851
A-n61-k09	—	—	—	0	0	100	1	0	3855	916	1.00	25897
A-n62-k08	—	—	—	2	17	83	4	1	1430	1227	1.00	20997
A-n63-k09	1.00	1.00	2418	2	50	50	5	2	641	1862	1.00	7742
A-n63-k10	1.00	1.00	6898	2	14	86	5	2	2539	1391	1.00	6811
A-n64-k09	—	—	—	1	34	66	2	0	2364	837	1.00	7550
A-n65-k09	—	—	—	2	81	19	4	0	5531	774	—	—
A-n69-k09	1.00	1.00	4388	2	24	76	5	2	1536	704	1.00	4180
A-n80-k10	—	—	—	0	0	100	1	0	1455	545	—	—

Table C.4: Results on JNVRGOC instances with 5 players and 3 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.81	0.65	263	10	89	11	14	5	342	254	0.74	183
A-n33-k06	1.00	1.00	280	4	34	66	9	4	1581	371	1.00	496
A-n34-k05	1.00	1.00	1381	9	83	17	14	4	2200	772	1.00	4269
A-n36-k05	0.91	x	942	10	48	52	15	5	786	606	0.89	32152
A-n37-k05	0.93	0.93	3659	11	39	61	17	5	2143	880	0.93	5297
A-n37-k06	0.65	x	336	7	69	31	13	5	571	468	0.44	588
A-n38-k05	1.00	1.00	898	5	24	76	10	4	1590	480	1.00	2263
A-n39-k05	1.00	1.00	2193	6	47	53	11	4	995	1171	1.00	11949
A-n39-k06	0.68	x	1009	9	70	30	15	5	693	762	0.54	1320
A-n44-k06	1.00	1.00	1190	7	56	44	12	4	507	580	1.00	2892
A-n45-k06	1.00	1.00	586	10	85	15	14	4	406	476	1.00	514
A-n45-k07	1.00	1.00	6088	4	21	79	9	4	4922	1449	1.00	11912
A-n46-k07	—	—	—	6	92	8	7	0	1615	2040	1.00	33196
A-n48-k07	—	—	—	7	59	41	8	0	2420	2149	—	—
A-n53-k07	—	—	—	4	36	64	8	3	2081	1733	—	—
A-n54-k07	—	—	—	0	0	100	1	0	3350	956	—	—
A-n55-k09	—	—	—	1	18	82	3	0	6057	616	—	—
A-n60-k09	—	—	—	5	28	68	6	0	3613	1495	—	—
A-n61-k09	—	—	—	4	18	82	8	2	3521	1640	—	—
A-n62-k08	—	—	—	4	62	36	5	0	339	2366	—	—
A-n63-k09	—	—	—	0	0	100	1	0	2837	1414	—	—
A-n63-k10	—	—	—	4	26	74	6	0	4403	1331	—	—
A-n64-k09	—	—	—	0	0	100	1	0	2853	896	—	—
A-n65-k09	—	—	—	5	73	27	7	1	1225	2828	—	—
A-n69-k09	—	—	—	0	0	100	1	0	4014	1088	—	—
A-n80-k10	—	—	—	0	0	100	1	0	2257	590	—	—

Table C.5: Results on JNVRGOC instances with 5 players and 5 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.73	0.22	152	13	90	10	16	5	278	226	0.63	71
A-n33-k06	0.80	0.80	495	5	28	72	11	5	2096	519	0.80	800
A-n34-k05	0.71	x	3550	10	44	56	16	5	3598	1264	0.34	7697
A-n36-k05	—	—	—	3	11	89	5	0	9472	573	0.69	26227
A-n37-k05	0.76	0.62	1567	11	66	34	14	6	861	492	0.76	2526
A-n37-k06	0.84	0.78	604	11	64	36	17	6	646	547	0.81	1150
A-n38-k05	0.86	x	1444	9	53	47	15	5	1942	470	0.81	2464
A-n39-k05	0.76	x	2255	9	57	43	13	5	863	953	0.69	9228
A-n39-k06	0.76	x	1751	11	80	19	17	6	1123	755	0.62	1283
A-n44-k06	0.88	0.88	800	6	49	51	12	5	911	714	0.88	4346
A-n45-k06	0.38	x	2366	6	42	58	12	5	2197	716	0.18	2296
A-n45-k07	1.00	1.00	1050	5	47	53	10	4	1030	827	1.00	7271
A-n46-k07	—	—	—	2	63	37	4	0	3978	997	—	—
A-n48-k07	—	—	—	9	97	3	10	0	1685	1537	—	—
A-n53-k07	—	—	—	8	76	24	9	0	1435	1526	—	—
A-n54-k07	—	—	—	0	0	100	1	0	3431	875	—	—
A-n55-k09	1.00	1.00	7044	6	55	45	11	4	4480	1784	1.00	27511
A-n60-k09	—	—	—	5	76	24	6	0	1318	2634	—	—
A-n61-k09	1.00	1.00	1757	4	60	40	9	4	482	1629	1.00	11056
A-n62-k08	—	—	—	0	0	100	1	0	2133	1133	—	—
A-n63-k09	—	—	—	0	0	100	1	0	3046	795	—	—
A-n63-k10	—	—	—	0	36	64	2	0	4176	1040	—	—
A-n64-k09	—	—	—	0	0	100	1	0	3313	1037	—	—
A-n65-k09	—	—	—	8	78	22	9	0	1301	1969	—	—
A-n69-k09	—	—	—	0	0	100	1	0	3434	1136	—	—
A-n80-k10	—	—	—	0	0	100	1	0	2138	594	—	—

Table C.6: Results on JNVRGOC instances with 5 players and 7 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.69	x	208	14	86	14	17	5	253	238	0.63	120
A-n33-k06	0.81	0.26	239	10	62	38	13	5	1009	470	0.67	395
A-n34-k05	0.80	0.27	3145	14	40	60	18	5	3986	1002	0.70	4361
A-n36-k05	—	—	—	3	22	78	5	0	9708	527	0.59	33251
A-n37-k05	0.71	x	1171	12	44	56	18	5	664	520	0.70	2487
A-n37-k06	0.44	x	2785	6	29	71	12	5	2781	1467	0.36	4486
A-n38-k05	0.86	0.79	1648	10	59	41	14	5	1805	493	0.83	2340
A-n39-k05	0.74	0.57	2641	13	72	28	16	5	937	970	0.71	11389
A-n39-k06	0.72	0.57	1003	15	62	38	19	6	1072	682	0.68	1842
A-n44-k06	1.00	1.00	1219	8	53	47	13	4	1049	752	1.00	2871
A-n45-k06	1.00	1.00	2189	9	44	56	14	4	1710	545	1.00	5342
A-n45-k07	—	—	—	0	0	100	1	0	5611	1032	—	—
A-n46-k07	0.74	x	4649	14	69	31	18	5	1458	1365	0.65	15269
A-n48-k07	0.74	—	5421	12	66	34	16	5	1143	1936	—	—
A-n53-k07	—	—	—	2	26	72	3	0	2591	1536	—	—
A-n54-k07	—	—	—	0	0	100	1	0	3323	931	—	—
A-n55-k09	—	—	—	10	66	34	14	3	2959	2090	—	—
A-n60-k09	—	—	—	8	66	34	10	0	1211	2549	—	—
A-n61-k09	—	—	—	0	0	100	1	0	3930	912	—	—
A-n62-k08	—	—	—	4	31	29	5	0	1191	1039	—	—
A-n63-k09	—	—	—	3	44	47	4	0	960	2565	—	—
A-n63-k10	—	—	—	0	0	100	1	0	4889	777	—	—
A-n64-k09	—	—	—	0	0	100	1	0	2202	1163	—	—
A-n65-k09	—	—	—	6	23	53	7	0	2155	2445	—	—
A-n69-k09	—	—	—	0	0	100	1	0	2004	975	—	—
A-n80-k10	—	—	—	0	0	100	1	0	2179	594	—	—

Table C.7: Results on JNVRGOC instances with 7 players and 3 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.64	0.56	709	19	42	58	26	7	558	686	0.63	18812
A-n33-k06	0.80	x	772	17	95	5	24	7	1526	958	0.67	6339
A-n34-k05	0.70	—	4121	21	36	64	28	7	9561	1069	—	—
A-n36-k05	0.78	—	5316	19	61	39	26	7	2141	1578	—	—
A-n37-k05	0.73	—	2888	17	73	27	24	7	1182	898	—	—
A-n37-k06	0.56	—	5842	14	61	39	22	7	3940	2409	—	—
A-n38-k05	0.76	x	2307	17	67	33	25	7	2132	1118	0.52	25322
A-n39-k05	—	—	—	13	60	33	14	0	3144	1626	—	—
A-n39-k06	0.63	—	2135	16	70	30	24	7	1532	1212	—	—
A-n44-k06	—	—	—	12	34	60	13	0	3601	2024	—	—
A-n45-k06	0.82	—	3790	16	74	26	24	7	1911	1592	—	—
A-n45-k07	—	—	—	0	0	100	1	0	1206	649	—	—
A-n46-k07	—	—	—	17	64	30	18	0	1636	3054	—	—
A-n48-k07	—	—	—	0	16	84	2	0	5113	973	—	—
A-n53-k07	—	—	—	0	11	89	2	0	5704	983	—	—
A-n54-k07	—	—	—	0	14	86	2	0	3078	1105	—	—
A-n55-k09	—	—	—	18	74	26	20	0	4417	1889	—	—
A-n60-k09	—	—	—	1	17	83	3	0	3381	1179	—	—
A-n61-k09	—	—	—	0	0	100	1	0	2845	1094	—	—
A-n62-k08	—	—	—	0	0	100	1	0	2383	1175	—	—
A-n63-k09	—	—	—	0	0	100	1	0	2501	1116	—	—
A-n63-k10	—	—	—	8	44	56	10	0	1504	2434	—	—
A-n64-k09	—	—	—	4	38	38	5	0	1418	1988	—	—
A-n65-k09	—	—	—	1	11	89	3	0	2546	1523	—	—
A-n69-k09	—	—	—	0	0	100	1	0	3615	806	—	—
A-n80-k10	—	—	—	0	0	100	1	0	2150	448	—	—

Table C.8: Results on JNVRGOC instances with 7 players and 5 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.64	0.53	902	19	57	43	27	7	900	715	0.57	16329
A-n33-k06	0.71	x	449	16	82	18	22	7	1028	872	0.64	6169
A-n34-k05	0.57	0.19	4039	22	39	61	30	7	7960	1184	0.47	26038
A-n36-k05	0.68	—	3857	21	67	33	28	7	2450	1297	—	—
A-n37-k05	0.55	—	5834	19	31	69	27	7	1434	765	—	—
A-n37-k06	0.71	—	5243	19	71	29	25	7	3347	1986	—	—
A-n38-k05	0.68	0.44	2317	23	59	41	30	7	1956	992	0.67	15266
A-n39-k05	—	—	—	7	55	35	8	0	3134	1894	—	—
A-n39-k06	0.67	—	4262	22	64	36	30	8	1948	1126	—	—
A-n44-k06	—	—	—	17	89	11	18	0	4049	1932	—	—
A-n45-k06	—	—	—	20	79	21	28	7	2936	2083	—	—
A-n45-k07	—	—	—	1	1	13	2	0	4757	1261	—	—
A-n46-k07	—	—	—	15	91	6	16	0	1469	2406	—	—
A-n48-k07	—	—	—	0	11	89	2	0	6077	823	—	—
A-n53-k07	—	—	—	2	26	74	4	0	2429	1706	—	—
A-n54-k07	—	—	—	0	0	100	1	0	3128	897	—	—
A-n55-k09	—	—	—	4	63	37	6	0	3287	1515	—	—
A-n60-k09	—	—	—	0	100	1	0	0	2040	1045	—	—
A-n61-k09	—	—	—	0	0	100	1	0	3069	924	—	—
A-n62-k08	—	—	—	0	13	87	2	0	366	1452	—	—
A-n63-k09	—	—	—	0	0	100	1	0	2505	1710	—	—
A-n63-k10	—	—	—	0	0	100	1	0	5294	733	—	—
A-n64-k09	—	—	—	0	0	100	1	0	2761	505	—	—
A-n65-k09	—	—	—	1	8	92	3	0	2555	1626	—	—
A-n69-k09	—	—	—	0	84	16	1	0	2027	690	—	—
A-n80-k10	—	—	—	0	39	61	2	0	1007	1864	—	—

Table C.9: Results on JNVRGOC instances with 7 players and 7 clusters of optional customers.

Instance	RG										ENUM ALL	
	α'	α	$T(s)$	I	$T_i(\%)$	$T_g(\%)$	BP	WP	BBN	MCC	α	$T(s)$
A-n32-k05	0.52	0.40	750	22	52	48	26	7	637	655	0.48	13040
A-n33-k06	0.69	x	409	19	82	18	26	7	872	947	0.66	6054
A-n34-k05	0.52	0.43	3968	22	27	73	29	8	7928	967	0.50	34934
A-n36-k05	0.59	—	6850	19	51	49	27	7	3377	1353	—	—
A-n37-k05	—	—	—	24	21	79	30	5	1573	1105	—	—
A-n37-k06	0.68	—	5228	21	75	25	25	7	4236	1779	—	—
A-n38-k05	0.66	0.55	4069	22	54	46	28	7	2306	911	0.61	21586
A-n39-k05	—	—	—	6	62	31	7	0	3273	1688	—	—
A-n39-k06	0.61	—	4924	23	53	47	31	7	1659	915	—	—
A-n44-k06	0.71	—	5982	21	55	45	26	7	1851	1513	—	—
A-n45-k06	0.67	—	4183	20	53	47	26	7	1378	1316	—	—
A-n45-k07	—	—	—	1	63	37	3	0	5413	1192	—	—
A-n46-k07	—	—	—	21	91	6	22	0	805	1915	—	—
A-n48-k07	—	—	—	0	21	79	2	0	6014	812	—	—
A-n53-k07	—	—	—	4	18	75	5	0	2167	1891	—	—
A-n54-k07	—	—	—	0	0	100	1	0	3580	876	—	—
A-n55-k09	—	—	—	2	21	79	4	0	6224	1126	—	—
A-n60-k09	—	—	—	4	13	6	5	0	1694	1604	—	—
A-n61-k09	—	—	—	0	0	100	1	0	3132	1123	—	—
A-n62-k08	—	—	—	0	34	66	1	0	1092	1507	—	—
A-n63-k09	—	—	—	5	73	27	7	0	1106	2973	—	—
A-n63-k10	—	—	—	0	0	100	1	0	5233	781	—	—
A-n64-k09	—	—	—	0	0	100	1	0	2741	909	—	—
A-n65-k09	—	—	—	2	20	80	4	0	2401	1853	—	—
A-n69-k09	—	—	—	4	62	18	5	0	1675	2146	—	—
A-n80-k10	—	—	—	0	0	100	1	0	1589	1184	—	—

References

- P. Augerat. Approche polyédrale du problème de tournées de véhicules. Institut National Polytechnique de Grenoble, France, 1995.
- R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, oct 2011.
- E. Ballot and F. Fontane. Reducing transportation CO2 emissions through pooling of supply networks: perspectives from a case study in French retail chains. *Production Planning & Control*, 21(6):640–650, Sept. 2010.
- L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946–985, 2019.
- B. Dai and H. Chen. Proportional egalitarian core solution for profit allocation games with an application to collaborative transportation planning. *European Journal of Industrial Engineering*, 9(1):53, 2015.
- J. Drechsel and A. Kimms. Computing core allocations in cooperative games with an application to cooperative procurement. *International Journal of Production Economics*, 128(1):310–321, nov 2010.
- S. Engevall, M. Göthe-Lundgren, and P. Värbrand. The traveling salesman game: An application of cost allocation in a gas and oil company. *Annals of Operations Research*, 82:453–472, 1998.
- S. Engevall, M. Göthe-Lundgren, and P. Värbrand. The heterogeneous vehicle-routing game. *Transportation Science*, 38(1):71–85, feb 2004.
- D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.
- P. C. Fishburn and H. O. Pollak. Fixed-route cost allocation. *The American Mathematical Monthly*, 90(6):366–378, jun 1983.
- M. Gansterer and R. F. Hartl. Collaborative vehicle routing: A survey. *European Journal of Operational Research*, 268(1):1–12, jul 2018.
- M. Guajardo and M. Rönnqvist. A review on cost allocation methods in collaborative transportation. *International Transactions in Operational Research*, 23(3):371–392, sep 2015.
- M. Göthe-Lundgren, K. Jörnsten, and P. Värbrand. On the nucleolus of the basic vehicle routing game. *Mathematical Programming*, 72(1):83–100, jan 1996.
- S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, chapter 2, pages 33–65. Springer US, Boston, MA, 2005.
- M. A. Krajewska, H. Kopfer, G. Laporte, S. Ropke, and G. Zaccour. Horizontal cooperation among freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society*, 59(11):1483–1491, nov 2008.
- G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.
- J. K. Lenstra and A. H. G. R. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.
- J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, jun 2004.
- R. Martinelli, D. Pecin, and M. Poggi. Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, 239(1):102–111, nov 2014.
- D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, jun 2016.
- D. Pecin, A. Pessoa, M. Poggi, E. Uchoa, and H. Santos. Limited memory rank-1 cuts for vehicle routing problems. *Operations Research Letters*, 45(3):206–209, may 2017.
- E. Pérez-Bernabeu, A. A. Juan, J. Faulin, and B. B. Barrios. Horizontal cooperation in road transportation: a case illustrating savings in distances and greenhouse gas emissions. *International Transactions in Operational Research*, 22(3):585–606, nov 2014.
- G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51(3):155–170, 2008.
- M. A. van Zon, R. Spliet, and W. van den Heuvel. The joint network vehicle routing game. *Econometric Institute Research Papers*, 2019.
- V. V. Zakharov and A. N. Shchegryaev. Stable cooperation in dynamic vehicle routing problems. *Automation and Remote Control*, 76(5):935–943, may 2015.