The Commodity-Split Multi-Compartment Capacitated Arc Routing Problem

Hani Zbib and Gilbert Laporte

PUBLISHED IN COMPUTERS & OPERATIONS RESEARCH
Volume 122
Pages 104994
Year 2020

The Commodity-Split Multi-Compartment Capacitated Arc Routing Problem

Hani Zbib[a,*], Gilbert Laporte[a]

[a] *Canada Research Chair in Distribution Management and CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada, H3T 2A7*

Abstract

The purpose of this paper is to develop a data-driven matheuristic for the Commodity-Split Multi-Compartment Capacitated Arc Routing Problem (CSMC-CARP). This problem arises in curbside waste collection, where there are different recyclable waste types called fractions. The CSMC-CARP is defined on an undirected graph with a limited heterogeneous fleet of multi-compartment vehicle types based at a depot, where each compartment's capacity can vary depending on the waste fraction assigned to it and on the compression factor of that fraction in that vehicle type. The aim is to determine a set of least-cost routes starting and ending at the depot, such that the demand of each edge for each waste fraction is collected exactly once by one vehicle, without violating the capacity of any compartment. The CSMC-CARP consists of three decision levels: selecting the number of vehicles of each type, assigning waste fractions to the compartments of each selected vehicle, and routing the vehicles. Our three-phase algorithm decomposes the problem into incomplete solution representations and heuristically solves one or more decision levels at a time. The first phase selects a subset of attractive compartment assignments from all assignments of all vehicle types. The second phase solves the CSMC-CARP with an unlimited fleet of the selected assignments. This is done by our C-split tour splitting algorithm, which can simultaneously split a giant tour of required edges into feasible routes while making decisions on the fractions that are collected by each route. The third phase selects the set of best routes servicing all fractions of all required edges without exceeding the number of vehicles available of each type. The algorithm is applied to real-life instances arising from recyclable waste collection operations in Denmark, with graph sizes up to 6,149 nodes and 3,797 required edges, the waste sorted in three to six fractions, and four to six vehicle types with one to four compartments. Computational results show that the generated solutions favor combining different fractions together in vehicles with higher numbers of compartments, and that the algorithm adapts well to the characteristics of the data, in terms of the graph, vehicle types, degree of sorting, and to skewness in demand among waste fractions.

Keywords: Arc routing; waste collection; commodity-split multi-compartment capacitated arc routing problem; matheuristic; data-driven.

*Corresponding author at: Canada Research Chair in Distribution Management and CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada, H3T 2A7.
Email address: hani.zbib@hec.ca (Hani Zbib)

1. Introduction

The purpose of this paper is to develop a data-driven matheuristic for the *Commodity-Split Multi-Compartment Capacitated Arc Routing Problem* (CSMC-CARP), a variant of the *Capacitated Arc Routing Problem* (CARP) (Golden & Wong, 1981) arising in the curbside collection of recyclable waste, among other applications. In curbside waste collection, different types of recyclables, called *fractions*, are collected from one or more bins located at the households. Typical fractions include general waste, organic waste, plastic, metal, glass, and paper.

Local authorities such as municipalities, towns, villages, or counties are responsible for the design and planning of waste collection at the tactical and operational levels (Ghiani et al., 2014). They make decisions on the organizational aspects of collection, the types and numbers of waste fractions to be collected, the types of collection vehicles and bins used, and undertake the capacity planning of the vehicles and bins (Bing et al., 2016). Most often, curbside collection is organized at the street level, where all households on the same street are serviced by the same vehicle. Such an operational decision is attractive from the citizens' point of view since it means that all neighboring households on the same street are serviced together, and is attractive for the vehicles' operator since servicing full street segments at a time is more convenient than servicing single households on different streets.

One possible way to model curbside waste collection is to use a node routing representation by considering each household as a node with a positive demand for at least one waste fraction. However, the organization of the collection at street level and the requirement of collecting all households on the same street contiguously by the same vehicle make an arc routing representation better suited to model the problem. In the arc routing representation, each street is modeled as a link and the demands of all households on that street are aggregated on the link. Moreover, by modeling the problem as an arc routing problem, the size of the graph is highly reduced compared with that of the node routing representation. For example, if the collection area included 100 streets with on average 10 households per street, the graph in the node routing representation would have 1,000 nodes needing collection, while the arc routing representation would have only 100 links needing collection.

Depending on the curbside collection policy followed by the municipalities, the underlying mathematical model varies. If they opt for a collection policy with single-compartment vehicles, the resulting problem is a CARP, which is solved independently for each waste fraction. Alternatively, if they opt for a collection policy with multi-compartment vehicles, the collection can be organized according to one of two strategies. The first strategy consists of collecting all the bins of a household by the same vehicle, the underlying problem being the *No-Split Multi-Compartment CARP* (NSMC-CARP). The second strategy, which is the one considered in this paper, consists of allowing different bins at one household to be collected by different vehicles, the underlying problem being the *Commodity-Split Multi-Compartment CARP* (CSMC-CARP). Further details on these problems and on the waste collection applications motivating them can be found in Kiilerich & Wøhlk (2018).

The former strategy is attractive from the citizens' point of view, where they are visited by only one vehicle at every collection period. However, with the goal of an increased degree of recycling of household waste, and considering the skewness in the amount of waste produced for each fraction,

collecting all bins of a household by the same vehicle would result in a poor skewed packing of the different compartments, and hence requiring a higher number of collection routes than by single compartments (Zbib & Wøhlk, 2019). Additionally, with highly ambitious sustainability goals of recycling up to seven fractions at the household, the vehicular technology that allows the collection of that many fractions in different compartments of the same vehicle with a compression mechanism for each compartment, does not yet exist, and even if it did, would require an even higher number of routes as the compartments would be relatively small. Moreover, most municipalities do not own a homogeneous fleet of collection vehicles, and end up purchasing a small number of new vehicles over time when needed. This leads to a heterogeneous vehicle fleet in the type and number of vehicles, and in the number of compartments of each vehicle.

Hence, our study of the CSMC-CARP is motivated by its ability to model a collection strategy that better organizes the collection of recyclables. It targets the issue of skewness of the amount of waste at the households by making better decisions on which fractions to be collected together, and hence better packing the vehicles' compartments. It also represents more realistically the composition of municipal waste collections fleets, while aiming at a good utilization of these fleets.

The CSMC-CARP is defined on an undirected graph. We consider a limited heterogeneous fleet of single- and multi-compartment vehicle types available at a depot, with a varying number of compartments. The aim is to compute a set of least-cost routes that start and end at the depot, so that the demand of a required edge for each waste fraction is collected exactly once by the compartments of one vehicle collecting that fraction in at least one of its compartments, without violating the compartment capacities of any vehicle, or violating the availability of each vehicle type. If an edge is serviced by a vehicle, then all the fractions collected on that edge have to be collected by that vehicle, and partial collection of fractions is not allowed. This is due to the fact that from the point of view of the operator, it is more convenient to collect all the fractions a vehicle is collecting in its compartments from all households on the same route. Moreover, a compression factor is associated with each waste fraction and each vehicle type. That is, depending on the fraction assigned to a vehicle compartment, that fraction is compressed by its specific factor, leading to a fraction-dependent compartment capacity after compression. The compression factor is dependent on the nature of the waste fraction, on its final processing purpose, and on the technological specifications of the compression mechanism in the vehicle. For example, glass should not be compressed too much as to avoid its breaking into shards, which would be problematic in its handling at the treatment facility, while general waste can be highly compressed since it is destined for incineration.

1.1. Literature review

While the CARP has been extensively studied (Hertz et al., 2000; Lacomme et al., 2004; Prins et al., 2009; Santos et al., 2010; Luiz Usberti et al., 2013; Bartolini et al., 2013; Prins, 2014; Belenguer et al., 2014; Muyldermans & Pang, 2014; Chen et al., 2016; Vidal, 2017; Wøhlk & Laporte, 2018), to the best of our knowledge, Muyldermans & Pang (2010a) are the only authors to have studied the CSMC-CARP. They considered an unlimited homogeneous fleet of vehicles with a predefined compartment capacity for each fraction needing collection, and allowed each fraction from the same edge to be collected by different vehicles. They used a guided local search heuristic with the aim

of comparing the routing cost of co-collection by multi-compartment vehicles to that of separate collection by single-compartment vehicles. Similarly, only a few papers have been devoted to the NSMC-CARP. Zbib (2019) applied a multi-move chain descent heuristic to large-scale real-life Danish instances This heuristic was used by Zbib & Wøhlk (2019) to conduct a comparative analysis of different curbside waste collection systems in Denmark.

The Commodity-Split Multi-Compartment Capacitated Vehicle Routing Problem (CSMC-CVRP), the node routing counterpart of the CSMC-CARP, has received more attention. The multi-compartment CVRP was first studied in the context of the distribution of gasoline (van der Bruggen et al., 1995; Avella et al., 2004). Most available solution methods are heuristics, due to the complexity of the CSMC-CVRP and the size of its solution space as opposed to that of the CVRP, even under a homogeneous unlimited fleet (El Fallahi et al., 2008; Wang et al., 2014). El Fallahi et al. (2008) considered a variant of the CSMC-CVRP in the context of the distribution of farm animal feed with an unlimited homogeneous fleet, the commodities being preassigned to compartments, but allowing for different commodities of the same node to be collected by different vehicles. They solved the problem through a memetic algorithm and tabu search based on the splitting of chromosomes into feasible routes. Muyldermans & Pang (2010b) studied the CSMC-CVRP with an unlimited homogeneous fleet of vehicles with a predefined compartment capacity for each fraction, and similarly to Muyldermans & Pang (2010a), again used a guided local search heuristic to solve the problem and compare co-collection with single collection. A mathematical model for the fixed fleet homogeneous fleet CSMC-CVRP was provided by Derigs et al. (2011), who used different metaheuristics to solve the problem. Wang et al. (2014) studied the CSMC-CVRP with a limited heterogeneous fleet, and used a reactive guided tabu search heuristic to solve it. Finally, a variant of the CSMC-CVRP with flexible compartment sizes was studied in the context of the collection of colored glass waste by Henke et al. (2015) and was solved by variable neighborhood search. Some works apply exact methods such as branch-and-price for the CSMC-CVRP (Mirzaei & Wøhlk, 2019), as well as branch-and-cut (Archetti et al., 2014) and branch-and-price-and-cut (Archetti et al., 2015) for the CSMC-CVRP with flexible compartment sizes and split deliveries. However, these algorithms can only solve small instances with up to 50 customers and four commodities to optimality, the largest instance containing only 100 customers.

To the best of our knowledge, neither the CSMC-CARP with compression factors and commodity-dependent compartment capacities nor its node routing counterpart have ever been investigated, and our work aims to fill this gap.

1.2. Scientific contribution and organization of this paper

The CSMC-CARP with a limited heterogeneous fleet and commodity-dependent compartment capacities consists of three decision levels: selecting the number of vehicles of each type to use in the solution, assigning waste fractions to the compartments of each selected vehicle, and routing the vehicles. Our solution strategy consists of solving the CSMC-CARP by tackling each of these three decision levels either separately, or two at a time in a three-phase algorithm.

The first phase selects a subset of attractive compartment assignments from all possible compartment assignments with sufficient capacities to cover the total demand of all required edges for

each waste fraction, where a compartment assignment is the assignment of waste fractions to the compartments of a vehicle type. The selection of this subset is driven by the instance data, notably the characteristics of the graph, the vehicle types, and the number of fractions.

The second phase is a routing phase that takes the selected assignments as an input and solves the CSMC-CARP with an unlimited vehicle fleet, where each selected assignment is considered as a new vehicle type with unlimited availability. This phase consists of iteratively generating an ordered tour of all required edges in the graph, which is split into feasible routes serviced by the different assignments, while ensuring that all waste fractions of all required edges are included in one and only one route. One of our main scientific contributions is the C-split tour splitting algorithm which decomposes a giant tour into feasible least-cost routes while determining the waste fractions assigned to each route.

Finally, the last phase takes as input the pool of all routes obtained in the routing phase, determines whether any of the assignments of each vehicle type can feasibly service each route, and chooses a subset of least-cost routes to collect all waste fractions of all required edges, while respecting the available number of each vehicle type.

The algorithm is run on the large-scale benchmark instances for the CSMC-CARP of Kiilerich & Wøhlk (2018), which are obtained from real-life waste collection data of six Danish counties. We consider graph sizes that vary between 26 and 6,149 nodes, and between 19 and 3,797 required edges, and a sorting of the waste in three, four, or six recyclable fractions. The types of vehicles vary between two to six, and the number of compartments varies between one and four.

The remainder of the paper is structured as follows: Section 2 formally describes the CSMC-CARP and the notation, Section 3 presents our solution strategy, Section 4 the computational experiments, and our conclusions follow in Section 5.

2. Formal problem description and notations

The CSMC-CARP is defined on an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. A specific node $v_0 \in \mathcal{N}$ serves as the depot. A traversal cost $c_e > 0$ is associated with every edge $e \in \mathcal{E}$, which is independent of whether the edge is being serviced or deadheaded. Let $\mathcal{F}$ be the set of waste fractions that need to be collected, with $|\mathcal{F}| > 1$ and $\mathcal{E}_r \subseteq \mathcal{E}$ being the set of required edges. We call the *degree of sorting* the number of fractions $|\mathcal{F}|$ that the waste is sorted in. For each edge $e \in \mathcal{E}_r$ and each waste fraction $f \in \mathcal{F}$ is associated a non-negative demand $q_e^f \geq 0$, with $\sum_{f \in \mathcal{F}} q_e^f > 0$. We also denote by $\mathcal{E}_r^f = \{e \in \mathcal{E}_r : q_e^f > 0\}$ the subset of required edges with a positive demand for waste fraction $f \in \mathcal{F}$. A limited heterogeneous fleet of multi-compartment vehicles are based at the depot. Let $\mathcal{K}$ be the set of vehicle types that form the fleet, and let b^k be the number of vehicles of each type $k \in \mathcal{K}$, each having a set $\mathcal{M}^k$ of compartments, with $|\mathcal{M}^k| \leq |\mathcal{F}|$. We denote by $\bar{\mathcal{M}} = \max_{k \in \mathcal{K}} \left\{|\mathcal{M}^k|\right\}$ the maximum number of compartments over all vehicle types. Each compartment $m \in \mathcal{M}^k$ has a capacity Q^{mk}. With each waste fraction $f \in \mathcal{F}$ and each type $k \in \mathcal{K}$ is associated a compression factor γ^{fk}: if the waste fraction $f \in \mathcal{F}$ is collected by any of the compartments of a vehicle of type $k \in \mathcal{K}$, the total demand collected of f by the vehicle is

compressed by γ^{fk}. The parameter $Q^{fmk} = \gamma^{fk} Q^{mk}$ is referred to as the compressed capacity of $f \in \mathcal{F}$ if assigned to $m \in \mathcal{M}^k, k \in \mathcal{K}$ (i.e. the capacity after factoring in the compression factor).

The objective of the CSMC-CARP is to determine a set of least-cost routes that start and end at the depot, such that the totality of the demand of a required edge for each waste fraction is collected exactly once by the compartments of one vehicle collecting that fraction in at least one of its compartments, without violating the capacity of any compartment, or the number of available vehicles. While the problem allows the split of the fractions of an edge among different vehicles, all fractions collected by a vehicle servicing that edge have to be collected at the same time in that vehicle. Since $|\mathcal{M}^k| \leq |\mathcal{F}|, \forall k \in \mathcal{K}$, and the number of vehicles is limited, the solution space includes decisions on the number of selected vehicles of each type, the assignment of fractions to the compartments of these vehicles, and the routing of the selected vehicles.

We use the term *compartment assignment* to refer to the assignment of a waste fraction $f \in \mathcal{F}$ to each compartment $m \in \mathcal{M}^k$ of vehicle type $k \in \mathcal{K}$ that respects the compressed capacities of the compartments. More formally, a compartment assignment s of a vehicle type $k \in \mathcal{K}$ is a vector of dimension $|\mathcal{M}^k|$ whose components are waste fractions. For example, $(1, 2, 1, 4)$ is a possible assignment of a vehicle $k \in \mathcal{K}$ with $|\mathcal{M}^k| = 4$, and $|\mathcal{F}| = 6$, where fraction 1 is collected in compartments 1 and 3, fraction 2 in compartment 2, and fraction 4 in compartment 4. Fractions $3, 5$, and 6 are not collected by this vehicle.

Note that if the same waste fraction is assigned to more than one compartment of the same vehicle, the total capacity of the vehicle for that waste fraction is considered to be the total capacity of the compartments collecting it. Moreover, while it is guaranteed that $q_e^f \leq \max_{k \in \mathcal{K}, m \in \mathcal{M}^k} \left\{ Q^{fmk} \right\}, \forall e \in \mathcal{E}_r, f \in \mathcal{F}$, there typically exists some edges whose demand for a certain fraction exceeds the compressed compartment capacity of one or more compartments of some vehicle types.

To simplify the presentation of our solution strategy, we have made two modeling assumptions that may not hold under all real-life conditions. First, even though we consider real-life data from six counties in Denmark ranging from rural to urban, for simplicity we model all instances of the problem on undirected graphs. This representation is highly appropriate for rural areas which mostly contain two-way streets, and which constitute most of Denmark. However, urban areas which often contain one-way streets are better represented with mixed graphs. Second, we assume that there is no distinction between the service cost and deadhead cost of an edge. This assumption holds when considering the cost as being the distance traveled by each vehicle. However, when considering service time versus deadhead time, it does not hold anymore since service time changes according to the number of waste fractions collected by the vehicle. In this case, the service time depends on the assignment of fractions to compartments, and varies according to the number of waste fractions collected by the vehicle. That is, the service time is calculated as the deadhead time plus the total time it would take to collect one fraction from all households on a street multiplied by the total number of waste fractions collected by the vehicle. Nevertheless, our solution strategy can easily be extended to handle mixed graphs and compartment-dependent service times. We briefly point out how this can be done wherever applicable in the remainder of this paper.

3. Solution strategy

As mentioned, the CSMC-CARP is characterized by three decision levels: selecting the number of vehicles of each type to use, assigning fractions to the compartments of the selected vehicles, and creating feasible routes for the selected assigned vehicles to collect the different waste fractions of all required edges.

In the presence of one waste fraction ($|F| = 1$), the CSMC-CARP reduces to the Heterogeneous Fixed Fleet Arc Routing Problem, which is the arc routing counterpart of the Heterogeneous Fixed Fleet Vehicle Routing Problem (HFFVRP), introduced by Taillard (1999). The HFFVRP is NP-hard and has a higher computational complexity than the CVRP due to the fact that the solution space is much more constrained, giving rise to more infeasibilities during the search space (Taillard, 1999; Prins, 2009; Koç et al., 2016). For example, Prins (2009) presented a tour-splitting algorithm for the HFFVRP whose best time complexity is $\mathcal{O}\left(n|\mathcal{K}||\mathcal{E}_r|^{|\mathcal{K}|}\right)$, where n is the number of arcs in the auxiliary splitting graph. This time complexity is much higher than that of the tour splitting algorithm proposed for the CVRP or for the CARP (Prins et al., 2009), whose time complexity is $\mathcal{O}\left(|\mathcal{E}_r|^2\right)$. Moreover, the author also mentions that the choice of the initial giant tour to be split in the HFFVRP plays a much more important role than in the CVRP, where some giant tours may not be feasibly split using the available fleet. This is due to the fact that splitting the giant tour results in a set of routes of which some are poorly packed. Hence the number of routes needed to cover all edges may exceed the number of vehicles available.

Since the CSMC-CARP considers both commodity-split multi-compartment routing and a heterogeneous limited fleet, its complexity and solution space are much larger than those of the CSMC-CARP with an unlimited fleet, and of the HFFARP studied separately, making any solution approach that simultaneously tackles its different decision levels combinatorially prohibitive. One solution strategy is to decompose the problem into incomplete solution representations, which are then used to heuristically solve one or more decision sets at a time. This helps reduce the scope of the heuristic search to the solution space and hence reduce the complexity of the different subproblems yielded by the decomposition.

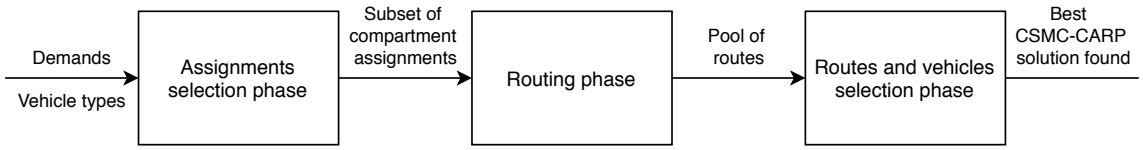

Figure 1: Overview of the algorithm.

We decompose the CSMC-CARP into three subproblems that are solved sequentially in a three-phase heuristic in which part or all of the solutions of the subproblem solved in a given phase constitute the input of the subsequent phase (see Fig. 1). The *assignments selection phase* takes as input the demands of all required edges for all waste fractions, and the number available of each vehicle type. It iteratively solves a *selection of compartment assignments subproblem* which consists of selecting, among all possible compartment assignments of all vehicle types, a diversified subset of assignments that can service the total demand for each waste fraction, while not exceeding the

availability of each vehicle type. The output of this phase is a subset of attractive compartment assignments that is driven by the characteristics of the graph, the vehicle types, and the number of fractions. This subset is then given as an input to the *routing phase*, which iteratively solves a *CSMC-CARP with an unlimited fleet of heterogeneous vehicle types*, where each compartment assignment obtained in the first phase is treated as a new vehicle type with an unbounded number of vehicles available. The output of each iteration is a full feasible solution of the subproblem, with the routes in the solution added to a pool of routes. This pool is then given to the *routes and vehicles selection phase*. This final phase determines for each route in the pool whether more than one compartment assignment of any vehicle type is able to service it, and gives the set of route-vehicle type pairs as an input to a *set partitioning subproblem*. The objective of this subproblem is to determine a least-cost subset of routes that collect each waste fraction of each required edge, while not exceeding the available number of each vehicle type. The final solution of the set partitioning subproblem corresponds to the best found CSMC-CARP solution.

The remainder of this section presents the details of the three-phases. Section 3.1 describes the assignments selection phase, Section 3.2 describes the routing phase, while Section 3.3 describes the routes and vehicles selection phase.

3.1. Assignments selection phase

The rationale behind the assignments selection phase is to choose a subset of attractive compartment assignments among the large number of possible assignments for each vehicle type. Given a vehicle type $k \in \mathcal{K}$ with $|\mathcal{M}^k|$ compartments, the total number of possible assignments can be obtained by computing $|\mathcal{M}^k|$ permutations with repetition out of $|\mathcal{F}|$ waste fractions, which corresponds to $|\mathcal{F}|^{|\mathcal{M}^k|}$. While this number is manageable for a degree of sorting of $|\mathcal{F}| = 3$, and $|\mathcal{M}^k| = 3$ (27 assignments), the number of possible assignments becomes way too large to consider entirely when $|\mathcal{F}| = 6$ and $|\mathcal{M}^k| = 4$ (1,296 assignments). In fact, for the largest vehicle data instance with $|\mathcal{K}| = 6$, $|\mathcal{F}| = 6$, and $\bar{\mathcal{M}} = 4$, the number of possible vehicle type-compartment assignment combinations is $\sum_{k \in \mathcal{K}} |\mathcal{F}|^{|\mathcal{M}^k|} = 1,806$ possible decisions (see Table 1). Therefore, only considering a smaller subset of compartment assignments yields a smaller decision space, while ensuring that the selected assignments are attractive.

An attractive subset of assignments is one that is diverse both in the combination of fractions serviced by each assignment and in its vehicle types. The subset is obtained by generating different subsets of assignments that minimize the number of vehicles used, while ensuring that the total compressed capacity of compartments collecting each waste fraction is at least as large as the total demand for that fraction, and the number of vehicles available of each type is not exceeded. The rationale for minimizing the total number of vehicles is that good solutions to the CSMC-CARP should favor a smaller number of vehicles used, and whose compartments have large compressed capacities capable of servicing a significant number of edges in the same route.

These subsets can be generated by solving the selection of compartment assignments subproblem. Here we present the notations and definitions of the subproblem, while (1)–(4) define its mathematical model:

S^k set of all possible assignments of waste fractions $f \in \mathcal{F}$ to the compartments $\mathcal{M}^k$ of vehicle type $k \in \mathcal{K}$, with $|S^k| = |\mathcal{F}|^{|\mathcal{M}^k|}$;

$\bar{S}$ final subset of attractive assignments, with $\bar{S} \subseteq \bigcup_{k \in \mathcal{K}} S^k$;

δ_s^k dummy cost associated with each assignment $s \in S^k, k \in \mathcal{K}$;

$$a_s^{fm} = \begin{cases} 1 & \text{if waste fraction } f \in \mathcal{F} \text{ is assigned to compartment } m \in \mathcal{M}^k \\ & \text{in assignment } s \in S^k; \\ 0 & \text{otherwise;} \end{cases}$$

Q_s^f total compressed capacity of waste fraction $f \in \mathcal{F}$ in assignment $s \in S^k$, $k \in \mathcal{K}$, with $Q_s^f = \sum_{m \in \mathcal{M}^k} a_s^{fm} Q^{fmk}$;

x_s^k non-negative integer variable corresponding to the number of selected vehicles of type $k \in \mathcal{K}$ with assignment $s \in S^k$.

$$\text{minimize} \quad \sum_{k \in \mathcal{K}} \sum_{s \in S^k} \delta_s^k x_s^k \tag{1}$$

$$\text{subject to} \quad \sum_{s \in S^k} x_s^k \leq b^k \qquad k \in \mathcal{K} \tag{2}$$

$$\sum_{k \in \mathcal{K}} \sum_{s \in S^k} Q_s^f x_s^k \geq \sum_{e \in \mathcal{E}_r^f} q_e^f \qquad f \in \mathcal{F} \tag{3}$$

$$x_s^k \geq 0 \text{ and integer} \qquad s \in S^k,\ k \in \mathcal{K}. \tag{4}$$

The objective function (1) minimizes the total cost of compartment assignments selected over the set of all possible assignments of all vehicle types. Constraints (2) are vehicle type constraints which ensure that the total number of vehicles of type $k \in \mathcal{K}$ selected does not exceed the total number b^k of vehicles available of that type. Constraints (3) are waste fraction constraints which ensure that the total compressed capacity of all compartments collecting fraction $f \in \mathcal{F}$ is sufficient to cover the total demand of all required edges $e \in \mathcal{E}_r^f$. Finally, constraints (4) define the domains of the variables. Table 1 presents the number of variables, number of vehicle constraints, and number of waste fraction constraints for the largest vehicle data instance for the different degrees of sorting into three, four, and six waste fractions respectively.

Table 1: Characteristics of the selection of compartment assignment subproblem for the largest vehicle file for different degrees of sorting.

Number of waste fractions	3	4	6
Number of variables	84	420	1,806
Number of vehicle constraints	6	6	6
Number of fraction constraints	3	4	6

Algorithm 1 presents the steps of the assignments selection phase, the output of the phase being the subset of attractive assignments $\bar{S} \subseteq \bigcup_{k \in \mathcal{K}} S^k$. The set $\bar{S}$ is obtained by iteratively solving the selection of assignments subproblem with updated dummy costs for a number of iterations β_0 (determined in a tuning phase) in order to obtain different subsets of assignments. Moreover, the

final constitution of $\bar{S}$ is data dependent, i.e. it is driven by the characteristics of the graph, the vehicle types, and the number of fractions.

Algorithm 1 Assignments selection phase.

Require: Q_s^f, q_e^f, γ^{fk}, $\forall s \in S^k, k \in \mathcal{K}, f \in \mathcal{F}, e \in \mathcal{E}_r$

```
Calculate γ̄^f and q̄^f,   ∀f ∈ F
H = ∅
for i = 1 to |F| − 1 do
    find {f ∈ F \ H : q̄^f = max_{h∈F\H} {q̄^h}}
    iterationCount = 0
    while iterationCount ≤ ⌊q̄^f β_0⌋ do
        Run the selection of compartment assignments subproblem on F \ H
        Add all s, x_s^k > 0 to S̄
        Update the dummy costs δ_s^k as in (7)–(8)
    end while
    H = H ∪ {f}
end for
return S̄
```

We start by calculating the average compression factor $\bar{\gamma}^f, \forall f \in \mathcal{F}$ over all vehicle types (eq. 5), and use the $\bar{\gamma}^f$ values to calculate an approximation of the average share $\bar{q}^f, \forall f \in \mathcal{F}$ that the total compressed demand for a waste fraction will occupy in the vehicles from the total compressed demand of all fractions (eq. 6). We use $\bar{q}^f$ in order to identify the waste fractions from most dominant in terms of total demand to least dominant. The subproblem is then solved for $\lfloor \bar{q}^f \beta_0 \rfloor$ iterations, after which the currently dominant fraction is removed from $\mathcal{F}$, the next dominant fraction is identified, and the subproblem is solved again for $\lfloor \bar{q}^f \beta_0 \rfloor$ iterations. This process is repeated until only two fractions remain. The rationale behind eliminating certain fractions in later iterations is that if a fraction highly dominates the others, then the subset of assignments obtained will include many assignments with that fraction, and very few assignments combining the other fractions. Moreover, intuitively speaking, those fractions that are more or less equally dominant are more likely to be paired together in a vehicle as the packing of the vehicle would be more balanced. However, in order to obtain a sufficiently large number of assignments with the more dominant fractions, we set the number of iterations for the solution of the subproblem proportional to the dominance of the fractions:

$$\bar{\gamma}^f = \frac{\sum_{k \in \mathcal{K}} \gamma^{fk}}{|\mathcal{K}|}, \quad f \in \mathcal{F} \tag{5}$$

$$\bar{q}^f = \frac{\sum_{e \in \mathcal{E}_r^f} q_e^f \bar{\gamma}^f}{\sum_{f \in \mathcal{F}} \sum_{e \in \mathcal{E}_r^f} q_e^f \bar{\gamma}^f}, \quad f \in \mathcal{F}. \tag{6}$$

The iterative process for the solution of the subproblem starts by considering all possible compartment assignments as being equally attractive (i.e. having a cost $\delta_s^k = 1$) in the first iteration, and penalizes some assignments in subsequent iterations by increasing their cost. At the end of each iteration, all $s \in \bigcup_{k \in \mathcal{K}} S^k, x_s^k > 0$ are added to the set $\bar{S}$. The cost penalization is done in

order to favor new subsets of compartment assignments that may initially be less attractive than the previously selected assignments, but are still attractive for the sake of diversification. Diversifying the types of vehicles selected as well as the combinations of fractions in the selected assignments is necessary due to the limitations imposed by the number of available vehicles of each type. With this aim in mind, the update of all the δ_s^k costs takes place under the following two conditions:

$$\text{if } x_u^k > 0, u \in \bar{S}, \text{ then } \delta_s^k = \delta_s^k + 1, s \notin \bar{S}, k \in \mathcal{K}; \tag{7}$$

$$\text{if } s \in \bar{S}, \text{ then } \delta_s^k = M, \text{ where } M \text{ is a large number.} \tag{8}$$

The cost update in (7) increases by one the cost of every assignment $s \notin \bar{S}$ that has not yet been selected for any assignment $u \in \bar{S}$ that has the same vehicle type. This ensures that if many assignments of the same vehicle type are selected, that vehicle type becomes less and less attractive, allowing for other vehicle types to be selected. The cost update in (8) sets the cost of already selected assignments $s \in \bar{S}$ to a sufficiently large value M, making them attractive under the sole condition that their reinclusion is needed to ensure feasibility at that iteration.

3.2. Routing phase

Even after obtaining the subset $\bar{S}$ of attractive compartment assignments, which leads to a reduced search space of the CSMC-CARP with a limited heterogeneous vehicle fleet, the search space is still too large to allow the simultaneous handling of the routing and the vehicle selection decisions. Therefore, in the routing phase, the vehicle availability constraints are relaxed, and routes are generated for the CSMC-CARP under the assumption that each of the assignments in $\bar{S}$ is a new vehicle type with unlimited availability. The aim of the routing phase is to iteratively create full solutions to the CMSC-CARP with an unlimited heterogeneous fleet of vehicle types corresponding to the compartment assignment in $\bar{S}$, by simultaneously assigning the required edges to the compartments of different assignments in $\bar{S}$, and creating routes for each used compartment assignment.

The general framework of the routing phase is given in Algorithm 2. Each iteration of this phase starts by creating a giant tour, which is an ordering of all the required edges $\mathcal{E}_r$ in the graph. We create giant tours using two different procedures, detailed in Section 3.2.2. The number of giant tours created β_1 is determined in the tuning phase. Each giant tour in $\mathcal{E}_r$ is then iteratively split into feasible routes at the waste fractions level (i.e. in the set $\mathcal{E}_r^f$) by the use of the C-split tour splitting algorithm (CSTSA), presented in Section 3.2.5. At every iteration of the CSTSA, the set of obtained routes is added to the pool $\mathcal{R}$ of routes identified so far, while ensuring that no duplicate routes exist in $\mathcal{R}$, and all routes in $\mathcal{R}$ are unique. For the sake of diversity, we apply a set of local search moves for the Rural Postman Problem (RPP), the switch and shorten moves of Hertz et al. (2000), on the giant tour, and then reapply the CSTSA on the post-moves giant tour.

This iterative process for each giant tour is bounded by two criteria: a maximum run time (in seconds) $\frac{60|\mathcal{E}_r|2^{|\mathcal{F}|}}{\beta_1\beta_2}$, which is dependent on the size of the graph and on the degree of sorting, and a maximum number of routes generated per giant tour $\alpha|\mathcal{F}|\frac{\beta_3}{\beta_1}$, which is dependent on the degree of

Algorithm 2 Routing phase.

Require: $\bar{S}$, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, F

```
for ω = 1 to β_1 do
    Create giant tour ω
    while runTime ≤ maxRunTime and numOfRoutes ≤ maxNumOfRoutes do
        Run the CSTSA on ω
        Apply a set of local search moves on ω
        Run the CSTSA on ω
    end while
end forreturn R
```

sorting and on the average number α of routes needed (where $\beta_1, \beta_2, \beta_3$ are parameters determined in a tuning phase). The value of α is obtained by considering the average number of vehicles needed to service all waste fractions in the assignments selection phase (i.e. in iteration $i = 1$ of lines 3–10 in Algorithm 1). The final output of the routing phase is the pool of all unique routes $\mathcal{R}$ obtained from all CSTSA iterations performed during the routing phase, without any knowledge of the vehicle type servicing each route.

3.2.1. Solution representation

The wide use of the tour splitting algorithm as subprocedures in several algorithms for the CARP and the CVRP is justified by the fact that once the subset of decisions on the order of edges is predetermined in a giant tour, the optimal splitting of that giant tour into feasible routes reduces to solving a shortest path problem on an auxiliary directed acyclic graph. Each edge in the giant tour is represented as a node in the auxiliary graph, and all feasible CARP subroutes capable of servicing subsequences of the edges in the order they appear in the giant tour are represented as arcs (Prins et al., 2009). However, in the case of the CSMC-CARP with a heterogeneous vehicle fleet, the solution representations of both the giant tour and its corresponding auxiliary graph are not as intuitive.

El Fallahi et al. (2008) present a tour splitting algorithm for the CSMC-CVRP with a homogeneous fleet of unlimited vehicles, where the assignment of fractions to compartments is fixed and known before hand. They encode the giant tour as an explicit ordering of customer node-fraction pairs (v, f), with $|F||\mathcal{N}|$ elements in the giant tour, $\mathcal{N}$ being the number of customer nodes. For example, if $|\mathcal{F}| = 2$ and $|\mathcal{N}| = 5$, then a giant tour has 10 node-fraction pairs, and one possible ordering is $((v_1, 1), (v_1, 2), (v_2, 1), (v_2, 2), (v_3, 1), (v_4, 1), (v_5, 1), (v_3, 2), (v_4, 2), (v_5, 2))$.

The corresponding auxiliary graph is a directed acyclic graph representing all feasible subroutes of the giant tour respecting the capacity of each compartment in the vehicle. Computing an optimal splitting of the giant tour reduces to solving a shortest path problem on the auxiliary graph, as for the CARP. While this solution representation is attractive with a homogeneous fleet with preassigned compartments, it is less attractive with a heterogeneous fleet of non-identical compartment assignments and given the fact that a vehicle has to collect from an edge all fractions it collects in its compartments. This pitfall is due to the encoding of the giant tour as an ordering of node-fraction pairs.

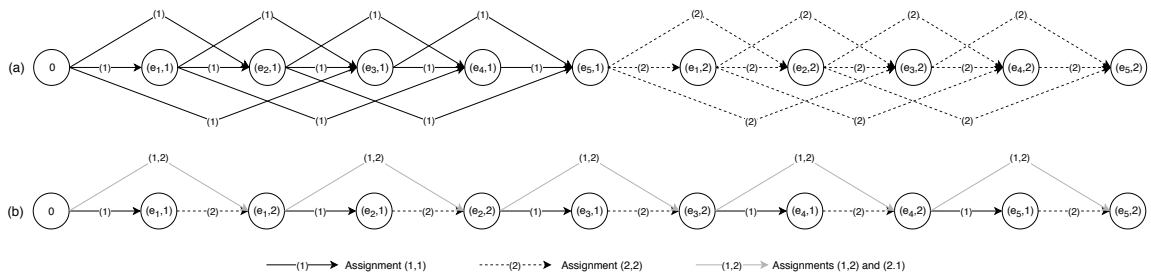

Figure 2: An example with $|\mathcal{F}| = 2$ and $|\mathcal{E}_r| = 5$ of the auxiliary splitting graph for the CSMC-CARP using the solution representation of El Fallahi et al. (2008).

For clarity of comparison with our problem, we consider in the following example the arc routing version of the problem of El Fallahi et al. (2008). In an arc routing setting, the giant tour is formed by all required edge-fraction pairs, the number of possible elements in the giant tour being $|\mathcal{F}||\mathcal{E}_r|$. Note that while the corresponding auxiliary graph is the same in both problems, the subtle difference in the giant tour representation between the arc routing version and the node routing version is that in the former case, there are two orientations of the edge of each edge-fraction pair, while in the latter case there is only one orientation for each node-fraction pair.

Given that $|\mathcal{F}| = 2$, $|\mathcal{E}_r| = 5$, one vehicle type with $|\mathcal{M}^k| = 2$, and the set of assignments $\bar{S} = \{(1,1), (2,2), (1,2), (2,1)\}$, one possible ordering of the 10 pairs is $((e_1,1),(e_2,1),(e_3,1),(e_4,1),(e_5,1),(e_1,2),(e_2,2),(e_3,2),(e_4,2),(e_5,2))$ (see Fig.2(a)). In this case, we can only form routes corresponding to assignment $(1,1)$ between the first five pairs, routes corresponding to assignment $(2,2)$ between the last five pairs, and no routes corresponding to assignments $(1,2)$ and $(2,1)$. On the other hand, if the ordering is $((e_1,1),(e_1,2),(e_2,1),(e_2,2),(e_3,1),(e_3,2),(e_4,1),(e_4,2),(e_5,1),(e_5,2))$ (see Fig.2(b)), then all possible routes corresponding to the assignments $(1,2)$ and $(2,1)$ can be formed between the two pairs of the same edge only, while routes corresponding to the assignments $(1,1)$ or $(2,2)$ can only be formed for every single edge-fraction pair. Therefore, the ordering of the fraction in the giant tour highly affects the routes that can be formed along the tour for different assignments, where a more mixed ordering favors the assignments with many fractions, and a more homogeneous ordering favors the assignments with few fractions.

In order to circumvent this problem, we consider an alternative solution representation. We encode the giant tour as an ordering of the edges in $\mathcal{E}_r$, and show that the optimal splitting of the giant tour into feasible routes reduces to solving a min-cost multi-commodity flow problem on an auxiliary directed acyclic multi-graph, which we define in Section 3.2.3. Note that the number of possible giant tours is $(|\mathcal{E}_r|)!$, which is significantly lower than the number $(|\mathcal{F}||\mathcal{E}_r|)!$ of possible giant tours in the solution representation of El Fallahi et al. (2008). Also note that our solution strategy is easily extendable to the CSMC-CARP defined on mixed graphs by considering one orientation in the giant tour for directed arcs and two orientations for undirected edges.

3.2.2. Giant tour creation

In order to create the giant tours as ordering of the required edges $\mathcal{E}_r$ in the graph, we use two different giant tour creation procedures. The first consists of using the well-known Frederickson

heuristic for the RPP (Frederickson, 1979) which gives us a good quality RPP giant tour.

However, as has been mentioned previously, the choice of the initial giant tour plays a more important role under a limited heterogeneous fleet of vehicles in the HFFVRP, as there could exist no feasible splitting of the giant tour into a set of routes that satisfy the availability of vehicle types. In fact, such a feature is more prominent in the CSMC-CARP with a limited heterogeneous fleet than in the HFFVRP as the packing component of the problem reduces to a multi-dimensional multi-commodity bin packing problem with commodity-dependent bin capacities and a limited number of bins, which is NP-hard and more difficult to solve than the basic bin packing problem.

Moreover in our experiments, the splitting of the giant tour led to an infeasible solution to the CSMC-CARP on instances whose characteristics present a clear dominance of a fraction over the others, as well as certain edges in proximity to each other with a very high demand for the dominant fraction as opposed to the other fractions (for further details on the specificity of the data, see Section 4.1). Therefore, in addition to generating the RPP tour by means of the Frederickson heuristic, we generate $\beta_1 - 1$ tours by applying a tour creation procedure that takes into account the delicate trade-off between the routing and the complex packing components of the problem. This procedure aims to generate a more diversified set of giant tours, leading to a diverse pool of routes, a subset of which are feasible to the CSMC-CARP. Algorithm 3 presents the giant tour creation procedure.

Algorithm 3 Giant tour creation procedure.

Require: $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, F

Calculate $\bar{q}_e, \quad \forall e \in \mathcal{E}_r$
current node $v \leftarrow v_0$
Giant tour $\omega \leftarrow \emptyset$
$\xi = \underset{e \in \mathcal{E}_r}{\text{average}} \{\bar{q}_e\}$
while $|\omega| < |\mathcal{E}_r|$ **do**
    if $\exists e \in \mathcal{E}_r, \notin \omega$ adjacent to v such that $\bar{q}_e \leq \xi$ **then**
        Add to ω at random any adjacent e to v such that $\bar{q}_e \leq \xi$
        $v \leftarrow$ second node in e
        $\xi = \underset{e \in \omega}{\text{average}} \{\bar{q}_e\}$
    else if $\exists e \in \mathcal{E}_r, \notin \omega$ adjacent to v such that $\bar{q}_e$ is minimal **then**
        Add e to ω
        $v \leftarrow$ second node in e
        $\xi = \underset{e \in \omega}{\text{average}} \{\bar{q}_e\}$
    else
        $v \leftarrow$ nearest node to v with at least one edge $e \in \mathcal{E}_r, \notin \omega$ adjacent to v
    end if
end while
return ω

The procedure starts by calculating for each edge $e \in \mathcal{E}_r$ the ratio $\bar{q}_e$ of the average compressed demand $q_e^f \bar{\gamma}^f$ for every fraction $f \in \mathcal{F}$ with a demand q_e^f, to the maximum compressed demand over all fractions (eq. 9). The value of $\bar{q}_e$ is both an indication of the skewness of the compressed demands among all fractions, and of the relative difference between the average compressed demand and the maximal compressed demand for the most dominant fraction. The rationale of the procedure

is to keep track of the average $\bar{q}_e$ of all edges added to the giant tour ω, and only adding (when possible) an edge e to ω which is nearest to the last edge added to ω such that $\bar{q}_e \leq \underset{e \in \omega}{\text{average}} \{\bar{q}_e\}$. The procedure concludes when all edges $e \in \mathcal{E}_r$ have been added to ω:

$$\bar{q}_e = \frac{\underset{f \in \mathcal{F}: q_e^f > 0}{\text{average}} \left\{ q_e^f \bar{\gamma}^f \right\}}{\underset{f \in \mathcal{F}: q_e^f > 0}{\max} \left\{ q_e^f \bar{\gamma}^f \right\}}. \tag{9}$$

3.2.3. Auxiliary graph

Let Φ be the set of all non-empty combinations (subsets) ϕ of possible waste fractions, such that $\phi \subseteq \mathcal{F}, \phi \in \Phi, |\phi| = 1, ..., |\mathcal{F}|$, with $|\Phi| = \sum_{i=1}^{|\mathcal{F}|} \binom{|\mathcal{F}|}{i} = 2^{|\mathcal{F}|} - 1$. For every assignment $s \in \bar{S}$, there exists a corresponding $\phi \in \Phi$. For example, the assignment $(1, 3, 1)$ corresponds to $\phi = \{1, 3\}$, and $(1, 3, 1, 3)$ also corresponds to $\phi = \{1, 3\}$. Conversely, we denote by $S_\phi \subseteq \bar{S}$ the subset of assignments in $\bar{S}$ that correspond to the same $\phi \in \Phi$, with $\bigcup_{\phi \in \Phi} S_\phi = \bar{S}, \bigcap_{\phi \in \Phi} S_\phi = \emptyset$. An upper bound on the number of combinations $\phi \in \Phi$ with a corresponding set $S_\phi \subseteq \bigcup_{k \in \mathcal{K}} S^k$ and a maximum number $\bar{\mathcal{M}}$ of compartments is $\sum_{i=1}^{\bar{\mathcal{M}}} \binom{\bar{\mathcal{M}}}{i} = 2^{\bar{\mathcal{M}}} - 1$. However, since the set $\bar{S}$ is only a subset of $\bigcup_{k \in \mathcal{K}} S^k$, then the actual number of unique combinations corresponding to all assignments in $\bar{S}$ will also be significantly smaller than $2^{\bar{\mathcal{M}}} - 1$.

Given a giant tour as an ordering $(1, ..., n)$ of the required edges in $\mathcal{E}_r$, we define $\mathcal{G}_\sigma = (\mathcal{N}_\sigma, \mathcal{A}_\sigma)$ as its corresponding auxiliary graph. $\mathcal{G}_\sigma$ is a directed acyclic multi-graph defined as follows. The set $\mathcal{N}_\sigma$ is an ordered set of nodes that contains a dummy node σ_0, as well as one node $\sigma_i, i = 1, ..., n$ for each edge given by the ordering $(1, ..., n)$. Each arc $(\sigma_i, \sigma_j)^\phi \in \mathcal{A}_\sigma, i < j$ corresponds to a route starting and ending at the depot node $v_0 \in \mathcal{N}$, and servicing the edges given by the ordering $(i + 1, ..., j)$, and also corresponds to a combination $\phi \in \Phi$ of waste fractions, where at least one compartment assignment $s \in S_\phi$ is able to feasibly collect the demands on the route for all fractions $f \in \phi$ without exceeding the compartment capacities $Q_s^f, \forall f \in \phi$. Figure 3 depicts an example of the auxiliary graph $\mathcal{G}_\sigma$ for a giant tour of size $n = 3$, $|\mathcal{F}| = 2$, $|\bar{S}| = 3$, the capacities of the three compartment assignments being $(Q_s^1, Q_s^2) = \{(11, 0), (0, 12), (8, 8)\}$. The demands (q_e^1, q_e^2) for the two fractions of each edge are indicated at the top of each node in the auxiliary graph, and all feasible arcs $(\sigma_i, \sigma_j)^\phi$ for every combination ϕ are represented, the cumulative demand(s) of each arc being indicated on it. Note that in the case where an edge has a demand $q_e^f = 0$ for a subset of the waste fractions in the graph (such as edge σ_3 in the example with $q_3^2 = 0$), for the sake of completeness the arc $(\sigma_2, \sigma_3)^{\{2\}}$ is included in the graph with a cumulative capacity for fraction 2 equal to 0. Such arcs are later dealt with in the post-optimization procedure in the third phase of our solution strategy.

With every arc $(\sigma_i, \sigma_j)^\phi$ is associated a cost π_{ij} corresponding to a least-cost route servicing the edges given by the ordering $(i + 1, ..., j)$. Two parallel arcs between the same nodes $\sigma_i, \sigma_j \in \mathcal{N}_\sigma$ have the same cost π_{ij}, as the route cost is independent of the combination ϕ associated with the arc $(\sigma_i, \sigma_j)^\phi$. Moreover, a binary $|F|$-dimensional vector a_{ij}^ϕ is associated with each arc $(\sigma_i, \sigma_j)^\phi$, such that $a_{ij}^{f\phi} = 1$ if $f \in \phi$, and $a_{ij}^{f\phi} = 0$ otherwise. Between any two nodes $\sigma_i, \sigma_j \in \mathcal{N}_\sigma$, there can be up to $2^{\bar{\mathcal{M}}} - 1$ parallel arcs $(\sigma_i, \sigma_j)^\phi$ each with a unique $\phi \in \Phi$, assuming that the route associated with $(\sigma_i, \sigma_j)^\phi$ is feasible for at least one $s \in S_\phi$, and that $\bar{S} = \bigcup_{k \in \mathcal{K}} S^k$. Hence, the maximum

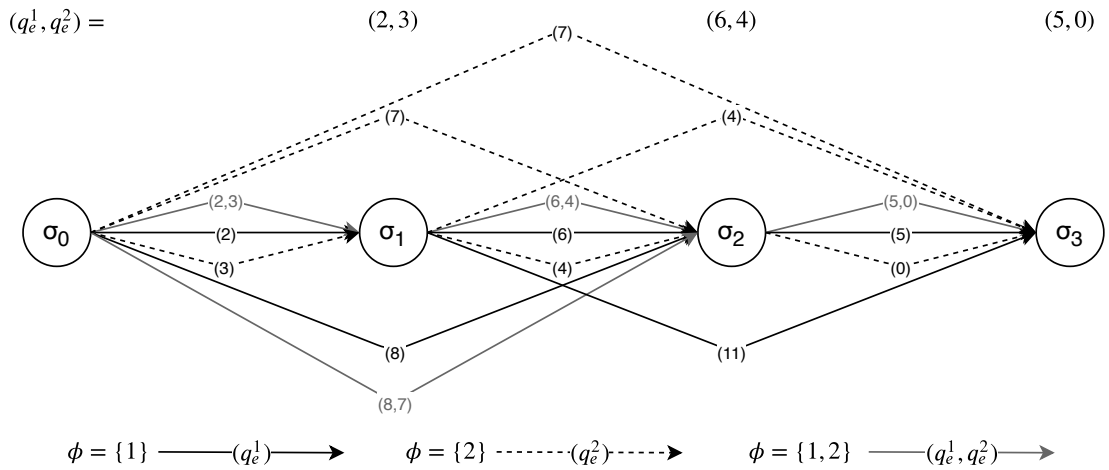

Figure 3: The auxiliary graph $\mathcal{G}_\sigma$ for a giant tour with $n = 3$, $|\mathcal{F}| = 2$, $|\bar{S}| = 3$, and $(Q_s^1, Q_s^2) = \{(11, 0), (0, 12), (8, 8)\}$.

number of arcs in the auxiliary graph is $(2^{\bar{\mathcal{M}}} - 1) \sum_{i=0}^{n-1} (n - i)$. In practice, the number of feasible arcs is smaller than $(2^{\bar{\mathcal{M}}} - 1) \sum_{i=0}^{n-1} (n - i)$. In the example of Figure 3, $\Phi = \{\{1\}, \{2\}, \{1, 2\}\}$, and therefore there could exist up to three parallel arcs between each two nodes in the auxiliary graph each corresponding to one ϕ (18 arcs in total). However, due to the capacity constraints, there exist only 15 feasible arcs in the graph.

3.2.4. Min-cost multi-commodity flow problem

A feasible solution of the CSMC-CARP in the auxiliary graph $\mathcal{G}_\sigma$ is a subset of arcs $(\sigma_i, \sigma_j)^\phi \in \mathcal{A}_\sigma$ such that a fully connected path can be followed from σ_0 to σ_n for each waste fraction $f \in \mathcal{F}$, and such that no two arcs overlap if they include the same waste fraction $f \in \mathcal{F}$ in their respective ϕ. For example, a feasible solution in Figure 3 is $\left\{(\sigma_0, \sigma_2)^{\{1\}}, (\sigma_0, \sigma_2)^{\{2\}}, (\sigma_2, \sigma_3)^{\{1,2\}}\right\}$.

Finding an optimal splitting of the giant tour over the sets $\mathcal{E}_r^f, \forall f \in \mathcal{F}$ into feasible routes serviced by the compartment assignments in $\bar{S}$ amounts to solving a min-cost multi-commodity flow problem on the auxiliary directed acyclic multi-graph $\mathcal{G}_\sigma = (\mathcal{N}_\sigma, \mathcal{A}_\sigma)$ with $|\mathcal{F}|$ commodities, where only one unit of flow of each commodity needs to be sent between the same source and sink nodes $\sigma_0, \sigma_n \in \mathcal{N}_\sigma$, respectively. Moreover, if a unit of flow for any $f \in \phi, |\phi| > 1$ is sent on arc $(\sigma_i, \sigma_j)^\phi$, then a unit of flow for each $f \in \phi$ also has to be sent along it. We present the mathematical model for this problem. The y_{ij}^ϕ variables are binary variables, where y_{ij}^ϕ equals 1 if the arc $(\sigma_i, \sigma_j)^\phi$ is being used, 0 otherwise:

$$\text{minimize} \quad \sum_{(\sigma_i,\sigma_j)^\phi \in \mathcal{A}_\sigma} \pi_{ij} y_{ij}^\phi \tag{10}$$

$$\text{subject to} \quad \sum_{\sigma_j \in \mathcal{N}_\sigma : (\sigma_0,\sigma_j)^\phi \in \mathcal{A}_\sigma} a_{0j}^{f\phi} y_{0j}^\phi = 1 \qquad f \in \mathcal{F} \tag{11}$$

$$\sum_{\sigma_i \in \mathcal{N}_\sigma : (\sigma_i,\sigma_n)^\phi \in \mathcal{A}_\sigma} a_{in}^{f\phi} y_{in}^\phi = 1 \qquad f \in \mathcal{F} \tag{12}$$

$$\sum_{\sigma_j \in \mathcal{N}_\sigma : (\sigma_i,\sigma_j)^\phi \in \mathcal{A}_\sigma} a_{ij}^{f\phi} y_{ij}^\phi - \sum_{\sigma_j \in \mathcal{N}_\sigma : (\sigma_j,\sigma_i)^\phi \in \mathcal{A}_\sigma} a_{ji}^{f\phi} y_{ji}^\phi = 0$$

$$\sigma_i \in \mathcal{N}_\sigma \setminus \{\sigma_0, \sigma_n\}, f \in \mathcal{F} \tag{13}$$

$$y_{ij}^\phi \in \{0,1\} \qquad f \in \mathcal{F},\ (\sigma_i,\sigma_j)^\phi \in \mathcal{A}_\sigma. \tag{14}$$

The objective function (10) minimizes the total cost of all arcs $(\sigma_i, \sigma_j)^\phi \in \mathcal{A}_\sigma$. Constraints (11) and (12) are flow balance constraints for each commodity $f \in \mathcal{F}$, respectively, for the first and last node $\sigma_0, \sigma_n \in \mathcal{N}_\sigma$, indicating that one unit of each commodity should leave σ_0, and one unit of each commodity should reach the last node σ_n. Constraints (13) are flow balance constraints for each commodity $f \in \mathcal{F}$ and for every node $\sigma_i \in \mathcal{N}_\sigma, \sigma_i \neq \sigma_0, \sigma_n$, ensuring that the same number of units of one commodity entering σ_i leave it. Finally, constraints (14) define the domains of the variables.

3.2.5. C-split tour splitting algorithm

The min-cost multi-commodity flow problem is NP-hard even if the auxiliary graph is an acyclic digraph and the source and sink nodes are the same for all commodities (Even et al., 1976). Unlike in the tour splitting algorithm for the CARP where a polynomial-time dynamic programming algorithm exists to solve the shortest path problem, no polynomial time algorithm is known to solve the min-cost multi-commodity flow problem. Therefore, we suggest a dynamic programming strategy based on labeling-setting to heuristically split the giant tour into feasible routes over the sets $\mathcal{E}_r^f, \forall f \in \mathcal{F}$ without creating the full auxiliary graph $\mathcal{G}_\sigma = (\mathcal{N}_\sigma, \mathcal{A}_\sigma)$. The algorithm is pseudo-polynomial in the number $|\mathcal{F}|$ of waste fractions. We denote by $g(z, \mathcal{F})$ a feasible solution to the min-cost multi-commodity flow problem, corresponding to a feasible split of the giant tour over the sets $\mathcal{E}_r^f, \forall f \in \mathcal{F}$.

The intuition behind the C-split tour splitting algorithm stems from the fact that any combination $\phi \in \Phi, |\phi| > 1$ can be obtained from the concatenation of a finite number of pairs of disjoint combinations also in Φ, whose cardinalities are smaller than $|\phi|$, and that form disjoint partitions of ϕ. That is, for every $\phi \in \Phi$, there exists a finite set $\Lambda(\phi)$ of pairs of combinations such that $\Lambda(\phi) = \Big\{ \{\phi_i, \phi_j\} : \phi_i, \phi_j \in \Phi, \phi_i \cup \phi_j = \phi, \phi_i \cap \phi_j = \emptyset \Big\}$. For example, the set $\Lambda(\phi)$ corresponding to the combination $\phi = \{1,2,3\}$ is $\Lambda(\phi) = \Big\{ \{1,2\} \cup \{3\}, \{1,3\} \cup \{2\}, \{2,3\} \cup \{1\} \Big\}$. Table 2 presents, for every $|\phi| = 2, ..., 6$, and for all possible combinations pairs $\{\phi_i, \phi_j\} \in \Lambda(\phi)$, the cardinality of ϕ_i and ϕ_j as $(|\phi_i|, |\phi_j|)$ and the total number of pairs $|\Lambda(\phi)|$. For example, in order to form a combination with cardinality $|\phi| = 3$ such as $\phi = \{1,2,3\}$, the pair of concatenated combinations can only have a cardinality $|\phi_i| = 2$ and $|\phi_j| = 1$, and there are three possible pairs that can be

concatenated together, i.e. $|\Lambda(\phi)| = 3$. On the other hand, forming a combination with cardinality $|\phi| = 4$ such as $\phi = \{1, 2, 3, 4\}$ can be done in two ways. The first way is to concatenate pairs with cardinality $|\phi_i| = 2$ and $|\phi_j| = 2$ such as $\{\phi_i, \phi_j\} = \{\{1, 2\}, \{3, 4\}\}$ or $\{\phi_i, \phi_j\} = \{\{1, 4\}, \{3, 2\}\}$, with a total of three possible concatenations. The second way is to concatenate pairs with cardinality $|\phi_i| = 3$ and $|\phi_j| = 1$ such as $\{\phi_i, \phi_j\} = \{\{1, 2, 3\}, \{4\}\}$ or $\{\phi_i, \phi_j\} = \{\{1, 2, 4\}, \{3\}\}$, with a total of four possible concatenations, giving that $|\Lambda(\phi)| = 7$.

Table 2: Characteristics of $\Lambda(\phi)$.

$\|\phi\|$	Number of fractions	$\|\Lambda(\phi)\|$
2	(1,1)	1
3	(1,2)	3
4	(1,3) (2,2)	7
5	(1,4) (2,3)	15
6	(1,5) (2,4) (3,3)	31

In the same spirit, a full solution $g(z, \mathcal{F})$ of the min-cost multi-commodity flow problem can be obtained from the concatenation of any pair of partial solutions $g(z, \phi_i), g(z, \phi_j) : \{\phi_i, \phi_j\} \in \Lambda(\mathcal{F})$, obtained from splitting the giant tour over all subsets $\mathcal{E}_r^f, \forall f \in \phi_i$ and $\mathcal{E}_r^f, \forall f \in \phi_j$. More generally, any solution $g(z, \phi), |\phi| > 1$ can be obtained from the concatenation of the partial solutions $g(z, \phi_i), g(z, \phi_j), \forall \{\phi_i, \phi_j\} \in \Lambda(\phi)$. Figure 4 shows one possible way of obtaining solution $g(z, \mathcal{F} = \{1, 2, 3, 4, 5, 6\})$ by recursively concatenating pairs of partial solutions $g(z, \phi_i), g(z, \phi_j)$. For example, the partial solution $g(z, \{2, 4\})$ can be obtained by solving the min-cost multi-commodity flow-problem for fractions 2 and 4 together, or as a concatenation of the problem solved for fraction 2 alone and fraction 4 alone. On the other hand, in order to obtain the solution $g(z, \{2, 3, 4, 5, 6\})$, the partial solution $g(z, \{2, 4\})$ can be concatenated with the partial solution $g(z, \{3, 5, 6\})$ in order to form a solution to the problem that includes fractions 2, 3, 4, 5, and 6. Finally, the full solution to the problem $g(z, \mathcal{F} = \{1, 2, 3, 4, 5, 6\})$ can be obtained by combining the solutions $g(z, \{2, 3, 4, 5, 6\})$ and $g(z, \{1\})$. Note that the example shows one way of obtaining $g(z, \mathcal{F})$, as there exist other pairs that could be concatenated together to form it, such as $g(z, \{1, 2, 3\})$ and $g(z, \{4, 5, 6\})$, for example.

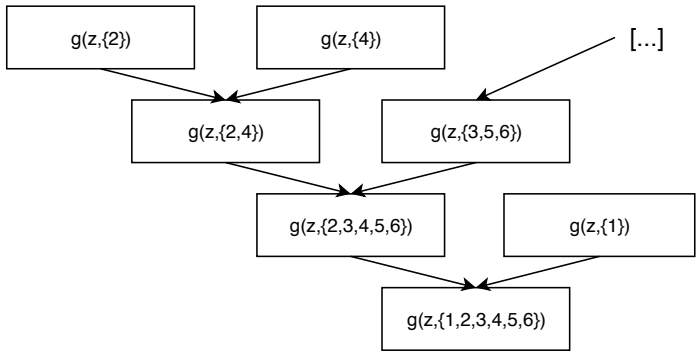

Figure 4: Example of a way to obtain $g(z, \mathcal{F} = \{1, 2, 3, 4, 5, 6\})$.

Given that the CSTSA is a label-setting algorithm, we define a set of cost labels L_i with cardinality $|\Phi|$ for every node $\sigma_i \in \mathcal{N}_\sigma$. For a given $t = 1, ..., |\Phi|$, there exists a one-to-one relationship between

the label $l_i^t \in L_i$ and every combination $\phi \in \Phi$, given by the function $\chi(t) = \phi$. We denote by $\chi^{-1}(\phi) = t$ the reverse function of χ. Table 3 presents the number of labels of size $|\chi(t)|$ for each of degree of sorting in three, four, and six waste fractions. For example, with $|\mathcal{F}| = 3$, each node requires seven labels in total, three labels for the three combinations $\{1\}$, $\{2\}$, and $\{1\}$ with cardinality $|\chi(t)| = 1$, three labels for the three combinations $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$ with cardinality $|\chi(t)| = 2$, and one label for the combination $\{1, 2, 3\}$ with cardinality $|\chi(t)| = 3$.

Table 3: Number of labels corresponding to combinations of sizes 1 to $|\mathcal{F}|$, for each degree of sorting.

$\|\chi(t)\|$	Number of fractions 3	4	6
1	3	4	6
2	3	6	15
3	1	4	20
4	-	1	15
5	-	-	6
6	-	-	1
$\|\Phi\|$	7	15	63

In order to both efficiently calculate the cost π_{ij} of routes in the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and evaluate the capacities of the routes, we precalculate partial distance and load labels for subsequences of edges in the ordering $(1, ..., n)$ as in Vidal (2016). For any subsequence of edges with ordering $(1, ..., i)$, D_i is the partial cost of the subsequence such that $D_i = c_1 + \sum_{j=2}^{i} (d_{j-1,j} + c_j)$, with $d_{j-1,j}$ being the shortest path cost between the end node of the edge at position $j - 1$ in the ordering and the start node of the edge at position j. Similarly, W_i^f is the partial load of the subsequence for fraction $f \in \mathcal{F}$, such that $W_i^f = \sum_{j=1}^{i} q_j^f, \forall f \in \mathcal{F}, \sigma_i \in \mathcal{N}_\sigma$. Finally, for any label $l_i^t, t = 1, ..., |\Phi|, \sigma_i \in \mathcal{N}_\sigma$, let $p(t, i)$ be the external predecessor node, $\psi(t, i)$ be the internal predecessor-pair of combinations $\{\phi_h, \phi_u\} \in \Lambda(\chi(t))$, and $\theta(t, i)$ be the cost predecessor-pair of combinations $\{\phi_h, \phi_u\} \in \Lambda(\chi(t))$.

The CSTSA is described in Algorithm 4. The first step is to precalculate in $\mathcal{O}(n)$ time the partial cost and load labels D_i and W_i^f, and initialize all the labels L_i and predecessors $p(t, i), \theta(t, i), \psi(t, i)$, $i = 0, ..., n, t = 1, ..., |\Phi|$. Then, looping through each node $\sigma_i \in \mathcal{N}_\sigma$, $i = 0, ..., n$ once, the CSTSA first updates the labels of $L_i, i \neq 0$ internally, and once all the labels of σ_i have been updated, it extends them to subsequent nodes σ_j, where $i < j \leq n$ and updates the labels of L_j. The internal label update procedure consists of determining for the subsequence $(1, ..., i)$ if there exists, for each partial solution $g(z, \phi), \forall \phi \in \Phi, |\phi| > 1, \chi^{-1}(\phi) = t$, a better label value for label l_i^t obtained from the concatenation of the pair of partial solutions $g(z, \phi_i), g(z, \phi_j), \forall \{\phi_i, \phi_j\} \in \Lambda(\phi)$. The label extension procedure for the labels L_j consists of determining for each $\phi \in \Phi$ whether the cost of the partial solution $g(z, \phi)$ for the subsequence $(1, ..., j)$ can be improved by reaching node σ_j from node σ_i.

Once the algorithm has looped through all the nodes in $\mathcal{N}_\sigma$, it outputs all partial solutions $g(z, \phi), \forall \phi \in \Phi$ obtained by the algorithm corresponding to the best found splitting of the giant tour into feasible routes servicing the subsets $\mathcal{E}_r^f, \forall f \in \phi$. This is done by outputting the labels $l_n^t, t = 1, ..., |\Phi|$ of the last node $\sigma_n \in \mathcal{N}_\sigma$, and the predecessors $p(t, n), \theta(t, n)$, and $\psi(t, n), t = 1, ..., |\Phi|$, which are then used to rebuild each partial solution $g(z, \phi), \chi^{-1}(\phi) = t$ and obtain the routes in that solution.

Algorithm 4 C-Split tour splitting algorithm.

Require: Φ, $\Lambda(\phi), S_\phi, \forall \phi \in \Phi$

Precalculate $D_i, W_i^f,\ i = 0, ..., n, \forall f \in \mathcal{F}$
Initialize the labels L_0 to 0
Initialize the labels $L_i,\ i = 1, ..., n$ to ∞
Initialize the predecessors $p(t,i), \theta(t,i), \psi(t,i)$ to $\emptyset$ for $i = 0, ..., n,\ t = 1, ..., |\Phi|$
for $i = 0$ to n **do**
    Update the internal labels $L_i, i \neq 0$
    Extend the labels L_i to the labels L_j of subsequent nodes with $i < j \leq n$
end for
return $l_n^t, p(t,n), \theta(t,n), \psi(t,n), t = 1, ..., |\Phi|$

The rationale behind including the routes from all partial solutions $g(z, \phi), \forall \phi \in \Phi$ is to diversify the pool of routes $\mathcal{R}$ in terms of fraction combinations due to the limited number of vehicles available with a larger number of compartments. Under an unlimited fleet of vehicles, the final $g(z, \mathcal{F})$ solution would always correspond to a feasible solution to split the giant tour. Such a solution would favor, when possible, servicing $\bar{\mathcal{M}}$ fractions at the same time in its routes. However, since the availability of vehicles with $\bar{\mathcal{M}}$ compartments is limited in our case, the obtained $g(z, \mathcal{F})$ solution would most probably be infeasible. Therefore, there is a need to include routes servicing one fraction up to $|\mathcal{F}|$ fractions, and considering all possible combinations of fractions that exist in $\bar{S}$.

Algorithm 5 details the internal label update procedure. For a given σ_i, the initial value of each $l_i^t \in L_i$ corresponds to the best found cost of the partial solution $g(z, \chi(t))$ for the ordering $(1, ..., i)$, by reaching σ_i from a preceding node. The procedure determines whether there exists a better solution $g(z, \chi(t)), |\chi(t)| > 1$ that can be obtained from the concatenation of pairs of solutions $g(z, \phi_h), g(z, \phi_u), \{\phi_h, \phi_u\} \in \Lambda_{\chi(t)}$. This is done by comparing the current label l_i^t to the the least-sum of the labels $l_i^{\chi^{-1}(\phi_h)} + l_i^{\chi^{-1}(\phi_u)}, \forall \{\phi_h, \phi_u\} \in \Lambda(\chi(t))$. If the sum is smaller than the value of the current label, this label updated with the sum, and the pair $\{\phi_h, \phi_u\}$ is set as the internal predecessor of label l_i^t. The labels are updated in increasing order of $|\chi(t)|$ in order to guarantee that all $l_i^{\chi^{-1}(\phi_h)}, l_i^{\chi^{-1}(\phi_u)}, \forall \{\phi_h, \phi_u\} \in \Lambda(\chi(t))$ have already been updated.

Algorithm 5 Internal label update procedure.

Require: i, L_i, Φ, $\Lambda(\chi(t))$

for $t = |\mathcal{F}| + 1$ to $|\Phi|$ **do**
    if $l_i^t > \min\limits_{\{\phi_h, \phi_u\} \in \Lambda(\chi(t))} \left\{ l_i^{\chi^{-1}(\phi_h)} + l_i^{\chi^{-1}(\phi_u)} \right\}$ **then**
        $l_i^t = \min\limits_{\{\phi_h, \phi_u\} \in \Lambda(\chi(t))} \left\{ l_i^{\chi^{-1}(\phi_h)} + l_i^{\chi^{-1}(\phi_u)} \right\}$
        $\psi(t,i) = \{\phi_h, \phi_u\}$
    end if
end for

Algorithm 6 details the extension of the labels L_i to the labels of subsequent nodes. The label extension loops through every node $\sigma_j, j = i + 1, ..., n$, and evaluates whether the value of each label l_j^t for $t = 1, ..., |\Phi|$ is larger than the value of the label l_i^t, plus the value of the cost function

$Cost(t,i,j)$ (eq. 15). If there exists at least one assignment $s \in S_{\chi(t)}$ for which the partial load difference $W_j^f - W_i^f$ does not exceed the compressed capacities $Q_s^f, \forall f \in \chi(t)$, then the route given by the ordering $(i+1,...,j)$ and servicing all $f \in \chi(t)$ is feasible, and the function outputs the route cost π_{ij}. However, if there exists no assignment $s \in S_{\chi(t)}$ for which the route is feasible, the cost function outputs the minimum cost among all combinations $\{\phi_h, \phi_u\} \in \Lambda(\chi(t))$ of feasible routes given by the ordering $(i+1,...,j)$ and servicing the fractions of the subsets $\phi_h, \phi_u \subset \chi(t)$. This is done by recursively calling the cost function for both $\chi^{-1}(\phi_h)$ and $\chi^{-1}(\phi_u)$ on i,j and summing their outputs. If $l_j^t > l_i^t + Cost(t,i,j)$, then the label l_j^t is updated, the node σ_i is set as the predecessor of the label t of node σ_j, and the cost predecessor-pair is updated if $Cost(t,i,j) > \pi_{ij}$. The procedure terminates prematurely at $j < n$ if no assignment $s \in S_{\chi(t)}$ for $t = 1,...,|\Phi|$ is feasibly able to service the route $(i+1,...,j)$.

$$Cost(t,i,j) = \begin{cases} \pi_{ij} = d_{v_0,i+1} + D_j - D_{i+1} + c_{i+1} + d_{j,v_0} & \exists\, s \in S_{\chi(t)} : W_j^f - W_i^f \leq Q_s^f, \forall f \in \chi(t) \\ \min\limits_{\{\phi_h,\phi_u\} \in \Lambda(\chi(t))} \left\{ Cost\left(\chi^{-1}(\phi_h), i, j\right) + Cost\left(\chi^{-1}(\phi_u), i, j\right) \right\} & \text{otherwise} \end{cases} \tag{15}$$

Finally, for the sake of diversifying the pool of routes $\mathcal{R}$ (looking ahead to the routes and vehicles selection phase), whenever the CSTSA is applied to split a giant tour, we run modified versions of the CSTSA subsequently in order to produce a pool $\mathcal{R}$ that is diverse in route sizes, compartment assignments, and in vehicle types. This is achieved by modifying the route feasibility criteria in the cost function $Cost(t,i,j)$ and (similarly line 8 of the label extension procedure (Alg. 6)) as follows, where η varies between 2 and $|\bar{\mathcal{M}}|$:

- $\exists\, s \in S_{\chi(t)} : W_j^f - W_i^f \leq Q_s^f, |\mathcal{M}^k| \leq \eta, \forall f \in \chi(t)$;
- $W_j^f - W_i^f \leq Q_s^f, \quad \forall s \in S_{\chi(t)}, |\mathcal{M}^k| \leq \eta, \forall f \in \chi(t)$.

Algorithm 6 Label extension procedure.

Require: i, L_i, Φ, $S_\phi, \forall \phi \in \Phi$
1: **for** $j = i+1$ to n **do**
2: **for** $t = 1$ to $|\Phi|$ **do**
3: **if** $l_j^t > l_i^t + Cost(t,i,j)$ **then**
4: $l_j^t = l_i^t + Cost(t,i,j)$
5: $p(t,j) = i$
6: **if** $Cost(t,i,j) > \pi_{ij}$ **then**
7: $\theta(t,j) \in \underset{\{\phi_h,\phi_u\} \in \Lambda(\chi(t))}{\arg\min} \left\{ Cost\left(\chi^{-1}(\phi_h), i, j\right) + Cost\left(\chi^{-1}(\phi_u), i, j\right) \right\}$
8: **end if**
9: **end if**
10: **end for**
11: **if** $\nexists\, s \in S_{\chi(t)} : W_j^f - W_i^f \leq Q_s^f, \forall f \in \chi(t),\ t = 1,...,|\Phi|$ **then**
12: Terminate the extension of the labels L_i
13: **end if**
14: **end for**

In order to determine the time complexity of the CSTSA, let $\overline{\Lambda} = \sum_{t=1}^{2^{|\mathcal{F}|}-1} |\Lambda(\chi(t))|$. The main loop of the CSTSA (lines 4–7 in Algorithm 4) iterates through the nodes of $\mathcal{N}_\sigma$ in $\mathcal{O}(n)$. One iteration of the internal label update procedure (Algorithm 5) runs in $\mathcal{O}\left(\overline{\Lambda}2^{|\mathcal{F}|}\right)$, while one iteration of the label extension procedure runs in $\mathcal{O}\left(n\overline{\Lambda}2^{|\mathcal{F}|}\right)$. This gives a total run time of $\mathcal{O}\left(n^2\overline{\Lambda}2^{|\mathcal{F}|}\right)$ for the CSTSA. Table 4 presents the value of $\overline{\Lambda}$ varying with the degree of sorting. Note that in practice, the run time of the label extension phase is not of the order of $\mathcal{O}(n)$, but of the order of the length of the largest feasible route for any $\phi \in \Phi$.

Note that our solution strategy easily extends to the CSMC-CARP with compartment-dependent service times by differentiating service cost and deadhead cost, defining a cost π_{ij}^{ϕ} in the auxiliary graph for every arc $(\sigma_i, \sigma_j)^{\phi}$ and combination ϕ, and extending the cost function $Cost(t, i, j)$ (eq. 15) to include service time.

Table 4: $\overline{\Lambda}$ for the different degrees of sorting.

t	Number of fractions 3	4	6
$\overline{\Lambda}$	6	25	301

3.2.6. Numerical example

To illustrate the algorithm, we use a numerical example with $|\mathcal{E}_r| = 5, |\mathcal{F}| = 3, \bar{\mathcal{M}} = 2$, and $\Phi = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\}$. Table 5 presents the numerical characteristics of each required edge and Table 6 the compressed capacities $Q_s^f, \forall f \in \phi, s \in S_\phi$. The best solution $g(z, \mathcal{F})$ found has a total cost of 127 and is represented on the auxiliary graph $\mathcal{G}_\sigma$ depicted in Figure 5, with the combination ϕ and the cost π_{ij} indicated on each arc $(\sigma_i, \sigma_j)^{\phi}$ in the solution. The final label values and all label predecessors for each $\sigma_i \in \mathcal{N}_\sigma$ are given in Table 7.

Table 5: Numerical characteristics of each edge.

Node	σ_1	σ_2	σ_3	σ_4	σ_5
q_i^1	5	1	2	4	2
q_i^2	4	1	3	6	5
q_i^3	3	2	5	4	1
c_i	3	4	2	5	7
$d_{v_0,i}$	6	5	3	5	9
$d_{i-1,i}$	4	7	3	8	6
d_{i,v_0}	10	2	4	6	5

Table 6: Compressed compartment capacities Q_s^f.

ϕ	S_ϕ
$\{1\}$	$\{(5, 0, 0), (6, 0, 0)\}$
$\{2\}$	$\{(0, 6, 0), (0, 8, 0)\}$
$\{3\}$	$\{(0, 0, 5)\}$
$\{1, 2\}$	$\{(4, 5, 0), (6, 6, 0)\}$
$\{1, 3\}$	$\{(5, 0, 6)\}$
$\{2, 3\}$	$\{(0, 6, 5)\}$

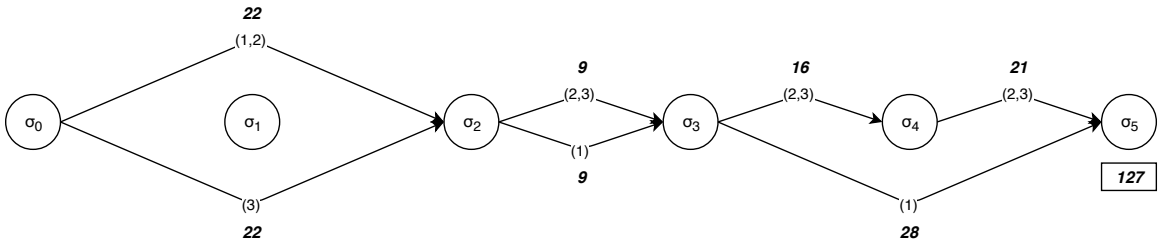

Figure 5: Final solution $g(z, \mathcal{F})$ of the numerical example represented on $\mathcal{G}_\sigma$.

Table 7: Numerical details of the CSTSA.

	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5
Label $\{1\}$	0	19	22	31	46	59
Label $\{2\}$	0	19	22	29	45	66
Label $\{3\}$	0	19	22	31	47	59
Label $\{1,2\}$	0	19	22	37	53	74
Label $\{1,3\}$	0	19	30	39	55	76
Label $\{2,3\}$	0	19	22	31	47	68
Label $\{1,2,3\}$	0	38	44	62	93	127
External predecessor $\{1\}$	-	0	0	2	2	3
External predecessor $\{2\}$	-	0	0	0	3	4
External predecessor $\{3\}$	-	0	0	2	3	3
External predecessor $\{1,2\}$	-	0	0	1	3	4
External predecessor $\{1,3\}$	-	0	1	2	3	4
External predecessor $\{2,3\}$	-	0	0	2	3	4
External predecessor $\{1,2,3\}$	-	0	0	2	3	4
Internal predecessor $\{1,2\}$	-	-	-	-	-	-
Internal predecessor $\{1,3\}$	-	-	-	-	-	-
Internal predecessor $\{2,3\}$	-	-	-	-	-	-
Internal predecessor $\{1,2,3\}$	-	-	-	-	$\{2,3\}$, $\{1\}$	$\{2,3\}$, $\{1\}$
Cost predecessor $\{1,2\}$	-	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
Cost predecessor $\{1,3\}$	-	$\{1,3\}$	$\{1,3\}$	$\{1,3\}$	$\{1,3\}$	$\{1,3\}$
Cost predecessor $\{2,3\}$	-	$\{2,3\}$	$\{2,3\}$,	$\{2,3\}$	$\{2,3\}$	$\{2,3\}$
Cost predecessor $\{1,2,3\}$	-	All comb.	$\{2,3\}$, $\{1\}$ or $\{1,2\}$, $\{3\}$	All comb.	$\{2,3\}$, $\{1\}$	All comb.

3.3. Routes and vehicles selection phase

After the pool of unique routes $\mathcal{R}$ has been obtained, it is given as an input to the routes and vehicles selection phase. The aim of this final phase is to find a subset $\mathcal{R}^* \subseteq \mathcal{R}$ of least-cost routes that service every waste fraction of every required edge, while satisfying the availability of each vehicle type. This is done by solving a set partitioning problem over the set of all required edge-waste fraction pairs. We present the notation and the mathematical model:

π_r cost of route $r \in \mathcal{R}$;

$$a_r^{ef} = \begin{cases} 1 & \text{if route } r \in \mathcal{R} \text{ services } f \in \mathcal{F} \text{ of edge } e \in \mathcal{E}_r^f; \\ 0 & \text{otherwise;} \end{cases}$$

λ_r^k binary variable. λ_r^k equals 1 if $r \in \mathcal{R}$ is collected by vehicle type $k \in \mathcal{K}$, 0 otherwise.

$$\text{minimize} \quad \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}} \pi_r \lambda_r^k \tag{16}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}} a_r^{ef} \lambda_r^k \geq 1 \qquad e \in \mathcal{E}_r^f, f \in \mathcal{F} \tag{17}$$

$$\sum_{r \in \mathcal{R}} \lambda_r^k \leq b^k \qquad k \in \mathcal{K} \tag{18}$$

$$\lambda_r^k \in \{0, 1\} \qquad k \in \mathcal{K},\ r \in \mathcal{R}. \tag{19}$$

The objective function (16) minimizes the total cost of all route-vehicle type pairs. Constraints (17) ensure that each waste fraction $f \in \mathcal{F}$ of a required edge $e \in \mathcal{E}_r^f$ is included in at least one route. Constraints (18) ensure that the total number of vehicles of each type $k \in \mathcal{K}$ does not exceed the total number available of that type. Constraints (19) define the domains of the variables.

Algorithm 7 details the routes and vehicles selection phase. The first step determines the set of vehicle types able to service each route. This is done by adding to the model, for each $r \in \mathcal{R}$, all variables λ_r^k corresponding to feasible route-vehicle type pairs r, k. Type k is paired with a route r associated with $\phi \in \Phi$ if there exists an assignment $s \in S^k, \bar{S}$ such that the total demands of $e \in r, \forall f \in \phi$ respect the compartment capacities of s. Once all feasible route-vehicle type pairs are found, the set partitioning model (16)–(19) is solved, returning the set of routes $\mathcal{R}^*$ of the best solution found.

Algorithm 7 Routes and vehicles selection phase.

Require: $\mathcal{R}$, $S_\phi, \forall \phi \in \Phi$
1: **for all** $r \in \mathcal{R}$ **do**
2: Add to model (16)–(19) variable $\lambda_r^k, \forall s \in S_\phi, S^k$ such that $\sum_{e \in r} q_e^f \leq Q_s^f, \forall f \in \phi$
3: **end for**
4: $\mathcal{R}^* \leftarrow$ best solution given by solving the set partitioning problem on all λ_r^k
5: Post-optimization of all routes $r \in \mathcal{R}^*$
6: **return** $\mathcal{R}^*$

Constraints (17) in the set partitioning model were relaxed by allowing an edge-fraction pair to appear in more than one route. This is due to the fact that we have not generated all possible routes, and there is no guarantee that the pool $\mathcal{R}$ includes a subset of routes perfectly covering all edge-fraction pairs. Moreover, even though every solution of the CSTSA is a full solution, there is also no guarantee that the routes of that solution can be assigned to vehicle types without violating the availability of each vehicle type. Therefore, the final step of the algorithm is to post-optimize the routes $r \in \mathcal{R}^*$. This is done by allowing one occurrence of each edge-fraction pair in the solution, and replacing all other occurrences by shortest path distances between the subsequent and consequent edges in the route. The same is undertaken for any edge $e \in r$ such that $q_e^f = 0, \forall f \in \phi$.

4. Computational experiments

We now describe the instances we have used to perform our tests, before presenting our results.

4.1. Data description

We have tested our algorithm on a subset of 63 CSMC-CARP instances taken from the set of benchmark instances for the CSMC-CARP of Kiilerich & Wøhlk (2018). Each instance consists of a graph, a degree of sorting, a set of vehicle types. The subset consists of 21 graphs, with three degrees of sorting in three, four, and six fractions (called B, D, and E in the instance files respectively), and each graph-degree of sorting pair coupled with one vehicle file. The smallest graph contains 26 nodes, 33 edges, and 19 required edges, and the largest contains 6,149 nodes, 7,110 edges, and 3,797 required edges. The vehicle files vary between four to six vehicle types, with one to four compartments, and 16 to 160 total number of vehicles available. The instances are available at http://www.optimization.dk/MC-CARP/.

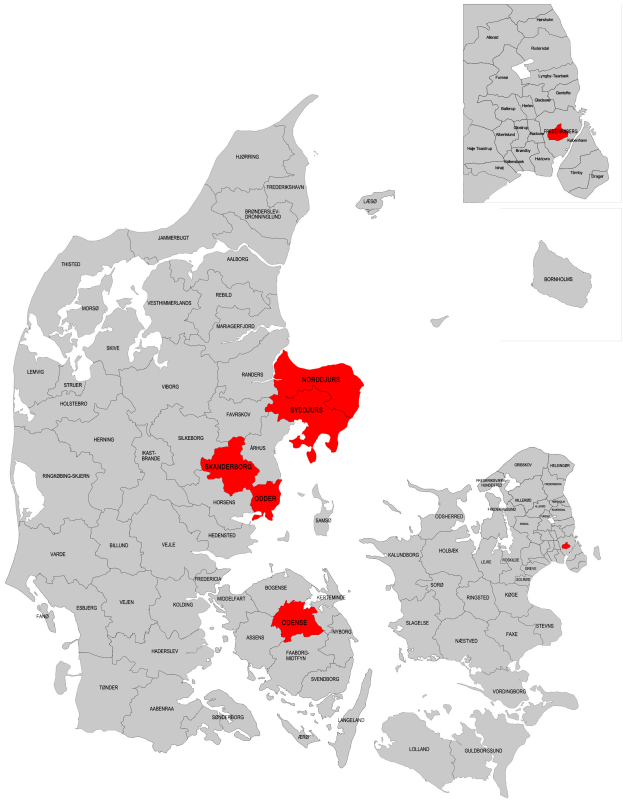

Figure 6: Map of the six Danish counties with a zoom on Frederiksberg in the top right corner.

The instances are obtained from a real-life application of curbside recyclable waste collection in Denmark. Table 8 provides the waste fractions of each degree of sorting at the source. A low degree of sorting means that several fractions are combined together at the point of collection, whereas a higher degree of sorting corresponds to a finer classification of the waste.

The graphs and waste demands are based on real-life road networks from six different counties and five municipal collaborators (see Fig. 6). The six counties are very dissimilar in their geographical and demographical characteristics: Norddjurs (N), Syddjurs (S), and the two counties of Skanderborg

and Odder (K), treated as one in our tests, are rural, Odense (O) is semi-urban, and Frederiksberg (F), located in central Copenhagen, is highly urban.

Table 8: Waste fractions composing each degree of sorting.

Degree of sorting	General	Organic	Plastic	Metal	Glass	Paper
3	General waste		Mixed resources			Paper
4	General waste	Organic waste	Mixed resources			Paper
6	General waste	Organic waste	Plastic	Metal	Glass	Paper

The dominance of the different waste fractions also varies among the different counties and the degree of sorting. Tables 9 to 11 present, for each of the three degrees of sorting respectively, the average per county of the average share from the total compressed capacity $\bar{q}^f$ for each waste fraction $f \in \mathcal{F}$. The different counties exhibit differences in the average share of the different fractions. For example, it is apparent that in the urban (F) county, general waste is highly dominant over the other fractions over all degrees of sorting. In contrast, the (N) county is the most balanced one, with paper having a larger share than in other counties. Finally, the highly urban nature of the Frederiksberg (F) instances (which is a commercially busy area of Copenhagen) showed a distinct specificity in the size of general waste demand relative to the compartment sizes of the vehicles. To illustrate, Figure 7 presents a box-plot comparison of the portion of a vehicle's total compartments capacity that the compressed demands for the general waste fraction would occupy in the vehicle, both for a graph from the semi-urban area of Odense (O) and Frederiksberg (F), compared with the same vehicle type. Most edges in the (O) graph occupy between 0% and 2% of the vehicle capacity, with only some edges with large demands occupying up to 20% of the capacity. In contrast, most of the edges in the (F) graph would occupy 4% to 20% of the vehicle capacity, with a few edges accounting for up to 40% and 54% of the vehicle capacity. Hence, the packing component of the CSMC-CARP is more predominant in the (F) graph than in the graphs of the other counties.

Table 9: Average shares $\bar{q}^f$ for a degree of sorting in three fractions.

County	General waste	Mixed Resources	Paper
F	70%	19%	11%
K	49%	41%	10%
N	40%	33%	27%
O	45%	37%	18%
S	47%	38%	15%

Table 10: Average shares $\bar{q}^f$ for a degree of sorting in four fractions.

County	General waste	Organic waste	Mixed resources	Paper
F	62%	10%	17%	10%
K	30%	33%	30%	7%
N	26%	28%	26%	21%
O	28%	30%	28%	13%
S	29%	31%	29%	12%

Table 11: Average shares $\bar{q}^f$ for a degree of sorting in five fractions.

County	General waste	Organic waste	Glass	Metal	Plastic	Paper
F	62%	10%	10%	4%	4%	10%
K	32%	35%	5%	5%	14%	8%
N	27%	29%	5%	5%	12%	23%
O	30%	32%	5%	5%	13%	14%
S	31%	33%	5%	5%	13%	12%

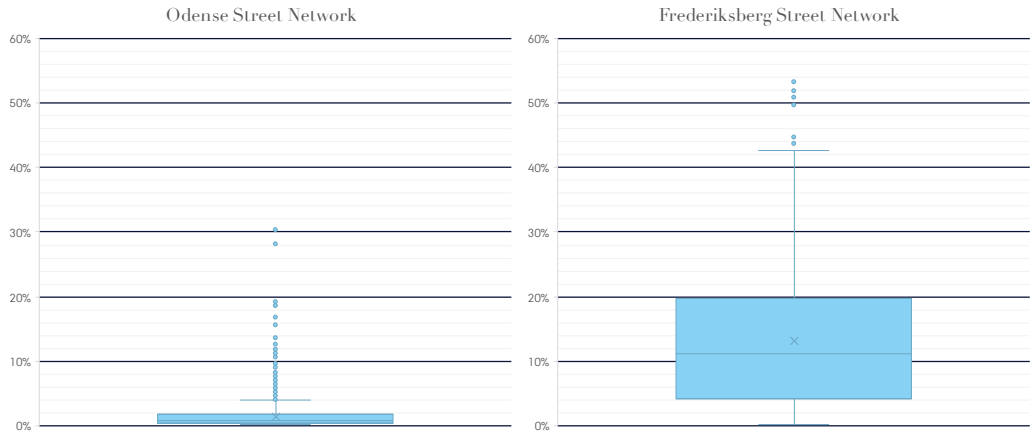

Figure 7: Portion of the vehicle that edges in the Odense (O) and Frederiksberg (F) networks would occupy for the general waste fraction.

4.2. Computational Results

The algorithm was implemented in C++ in MS Visual Studio Professional 2015, and we used CPLEX 12.9.0 to solve the mathematical models in the first and third phases. The algorithm was executed on a VMware virtual machine with the following specs: two Intel E5-2683 v4 Broadwell CPUs at 2.1Ghz with one core and 90GB of vRAM.

To tune the algorithm's parameters, we considered the number of iterations $\beta_0 = |\mathcal{F}|^2$ in the assignments selection phase. We also tried the values $|\mathcal{F}|$ and $|\mathcal{F}|^{\frac{3}{2}}$, but these proved to be too small to obtain a sufficient number of compartment assignments. On the other hand, a value of $2^{|\mathcal{F}|}$ did not yield sufficient improvements to justify the higher run time. We also limited the time of each CPLEX run of the selection of compartment assignments subproblem to 30 seconds, due to the fact that in later iterations the model was reaching an optimality gap of less than 1% within the first 30 seconds, but took a long time to reach optimality.

As for the parameters related to the routing phase (the number of giant tours β_1, the maximum run time β_2, and the maximum number of routes β_3 per iteration of the routing phase), the three parameters are related and affect each other. We have considered $\beta_1 = 10$ giant tours, which was a number of giant tours sufficiently high to allow a diversified pool of routes $\mathcal{R}$ while allowing enough run time and a number of routes generated from the splitting of each giant tour. As for the maximum run time, we took $\beta_2 = 533$, which allows a total run time for the routing phase of two hours for every 1,000 required edges, and a degree of sorting of six, corresponding to 12 minutes per giant

tour. This gives a maximum run time for the routing phase of $0.9|\mathcal{E}_r|$, $1.8|\mathcal{E}_r|$, and $7.2|\mathcal{E}_r|$ with a sorting degree in three, four, and six fractions, respectively. We have also taken $\beta_3 = 600$, thus allowing a total maximum number of routes for the routing phase of $1,800\alpha$, $2,400\alpha$, and $3,600\alpha$, respectively, for the three degrees of sorting, where α is the average number of vehicles needed to service all the waste fractions in the assignments selection phase. Finally, we capped the maximum number of routes to 100,000, as any number of routes higher than that was computationally heavy for the set partitioning subproblem in the routes and vehicles selection phase. In fact, we set a time limit of two hours on CPLEX for the set partitioning subproblem, which was enough to obtain the optimal solution to the subproblem in 65% of the instances, and a near-optimal solution in 35% of the instances with the worst optimality gap being 0.495%.

We first illustrate our algorithm in Tables 12 and 13 on two instances for a degree sorting $|\mathcal{F}| = 6$, with both instances sharing the same vehicle file M4-1 with $|\mathcal{K}| = 6$, $|\bar{\mathcal{M}}| = 4$, $\sum_{k\in\mathcal{K}} b^k = 40$, and the vehicles having, respectively, two, four, three, two, three, and one compartments. The first graph O12_E has $|\mathcal{N}| = 761$ and $|\mathcal{E}_r| = 533$, and the second graph F11_E has $|\mathcal{N}| = 191$ and $|\mathcal{E}_r| = 174$.

Table 12: Solution of graph O12_E and vehicle file M4-1.

Vehicle type	$\lvert\mathcal{M}^k\rvert$	Fraction combination	Compartments assignments	Number of routes	Route costs
0	2	$\{1,2\}$	$(1,2)$	3	19180, 27264, 27590
1	4	$\{3,6,5,4\}$	$(3,6,5,4)$	3	24793, 35336, 9591
1	4	$\{1,3,4\}$	$(1,3,1,4)$	2	49599, 22472
2	3	$\{2,5,6\}$	$(2,5,6)$	3	38621, 12272, 22472

Out of 1,806 possible assignments, 298 (16.5%) were chosen in the assignments selection phase for F11, and 191 (10.6%) for O12. The final solution for O12 contains 11 routes (28% of the total number of available vehicles) and has a cost of 289,190. It only uses the first three types of vehicles, with four distinct combinations, one compartments assignment for each combination, and multiple routes per assignment. On the other hand, the final solution for F11 contains 29 routes (73% of the total number of available vehicles) and has a cost of 78,919. It uses all six types of vehicles, and nine distinct combinations, some spanning multiple vehicle types and multiple vehicle assignments for the same vehicle type. This helps illustrate how the algorithm adapts to the characteristics of each graph, but also how the final solutions favored, when available, vehicles with many compartments over vehicles with fewer compartments.

Table A.1 in Appendix A provides the detailed results for each instance by reporting the best solution found, the total run time (in minutes) of the algorithm and its different phases, the number of routes in the solution, and the percentage of vehicles used from the total number available. Table A.2 provides further details on the algorithm by reporting, for each instance, the number of total assignments in the instance, the percentage of assignments chosen $\left(\frac{|\bar{S}|}{|\bigcup_{k\in\mathcal{K}} S^k|}\right)$, the total number of iterations the CSTSA was run, the size of the pool of routes $\mathcal{R}$, and the number of route-vehicle pairs given to the set partitioning model in the routes and vehicles selection phase.

In terms of computational run time, the algorithm ran in 0.25 second for the smallest graph and degree of sorting, and in 11.8 hours for the largest graph with 3,797 required edges and the largest

Table 13: Solution of graph F11_E and vehicle file M4-1.

Vehicle type	$\lvert\mathcal{M}^k\rvert$	Fraction combination	Compartments assignments	Number of routes	Route costs
0	2	$\{1\}$	$(1,1)$	4	1549, 4049, 2509, 2037
0	2	$\{2,6\}$	$(6,2)$	1	5753
1	4	$\{1,6\}$	$(1,6,1,1)$	1	1607
1	4	$\{1,2,6\}$	$(1,1,6,2)$ $(2,1,1,6)$ $(1,2,1,6)$	1 2 1	4311 2399, 3285 3682
2	3	$\{3,4,5\}$	$(4,3,5)$ $(4,5,3)$	1 2	3825 9463, 6738
2	3	$\{1,6\}$	$(1,6,1)$	1	1903
2	3	$\{1,2,6\}$	$(1,6,2)$	1	2106
3	2	$\{1\}$	$(1,1)$	1	1045
3	2	$\{2,3\}$	$(3,2)$	1	556
3	2	$\{2,6\}$	$(2,6)$	2	2351, 3279
3	2	$\{4,6\}$	$(4,6)$	1	2086
4	3	$\{1,4\}$	$(4,1,1)$	2	1471, 707
4	3	$\{1,6\}$	$(6,1,1)$	1	350
4	3	$\{3,5\}$	$(5,5,3)$	2	2086, 5223
5	1	$\{1\}$	(1)	4	1503, 969, 961, 1116

degree of sorting. The selection of assignments phase ran between 0.03 seconds and 15.8 minutes, the routing phase between 0.14 seconds and 9.3 hours, and the routes and vehicle selection phase between 0.04 seconds and 2.1 hours. The first phase took on average 5% of the computational time, the second phase 60%, and the third phase 35%. Moreover, every CSTSA iteration took on average one second over all instances, 0.0013 second for the smallest instance, and 14 seconds for the largest instance. The number of assignments selected in the assignments selection phase varied between 16 to 406, with an average of 118, and the percentage of assignments selected varied between 5% and 61%, with an average of 28%. Looking at the percentage of assignments selected for each degree of sorting, the average percentage of assignments selected decreases with an increased degree of sorting. On average, 47%, 23%, and 16% assignments were selected respectively for a degree of sorting in three, four, and six. As for the solution characteristics, The number of routes used varied between two and 130 vehicles, with on average 30 vehicles used. In terms of the utilization of vehicles from the total number available, the lowest utilization was 12%, the average 53%, and the largest utilization was 100%.

5. Conclusions

We have developed a data-driven matheuristic for the Commodity-Split Multi-Compartment Capacitated Arc Routing Problem with compression factors and a limited heterogeneous vehicle fleet. The problem is real and is motivated by the application of curbside recyclable waste collection from households. Due to the intricate combinatorial nature of the problem, which includes three different

decision levels, our algorithm decomposes the problem into incomplete solution representations and heuristically solves one or more decision levels at a time. We have introduced the C-split tour splitting algorithm, a novel algorithm that can simultaneously split a giant tour into feasible least-cost routes while making decisions on the waste fractions that are serviced by each route. The algorithm was tested on real-life waste collection instances from six counties in Denmark, exhibiting highly different features. The graphs contain up to 6,149 nodes, 7,110 edges, out of which 3,797 are required. We considered three degree of sorting in three, four, and six waste fractions, four to six vehicle types, with a number of compartments varying between one and four. The computational results have shown that the algorithm yields solutions that favor combining different fractions together in vehicles with a higher order of multiple compartment. Our algorithm also aptly adapts to the data, and is driven by the characteristics of the graph, the vehicle types, and the degree of sorting. Moreover, it handles well any skewness in the demands of different waste fractions, where one or more fractions dominate the others, be it in the total compressed demand share relative to the total compressed demand, or with respect to the edge demands in a given graph. Finally, our solution strategy can easily be extended to handle several variants of the CSMC-CARP, including an unlimited fleet of homogeneous or heterogeneous vehicles, mixed and directed graphs, a node routing setting, and a differentiation of the service cost and the deadhead cost.

Acknowledgment

This work was supported by the Danish Council for Independent Research - Social Sciences. Project "Transportation issues related to waste management" [grant number 4182-00021] and by the Canadian Natural Sciences and Engineering Research Council [grant number 2015-06189]. This support is gratefully acknowledged. We Thank Sanne Wøhlk for her valuable feedback. Thanks are due to the referees for their valuable comments.

Declaration of interest

None

References

Archetti, C., Bianchessi, N., & Speranza, M. G. (2015). A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. *Computers & Operations Research*, *64*, 1–10.

Archetti, C., Campbell, A. M., & Speranza, M. G. (2014). Multicommodity vs. single-commodity routing. *Transportation Science*, *50*, 461–472.

Avella, P., Boccia, M., & Sforza, A. (2004). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, *152*, 170–179.

Bartolini, E., Cordeau, J.-F., & Laporte, G. (2013). Improved lower bounds and exact algorithm for the capacitated arc routing problem. *Mathematical Programming*, *137*, 409–452.

Belenguer, J. M., Benavent, E., & Irnich, S. (2014). The capacitated arc routing problem: Exact algorithms. In Á. Corberán, & G. Laporte (Eds.), *Arc Routing: Problems, Methods, and Applications* chapter 9. (pp. 183–221). Philadelphia: MOS-SIAM Series on Optimization.

Bing, X., Bloemhof, J., Ramos, T., Barbosa-Povoa, A., Wong, C., & van der Vorst, J. (2016). Research challenges in municipal solid waste logistics management. *Waste Management*, *48*, 584–592.

Chen, Y., Hao, J.-K., & Glover, F. (2016). A hybrid metaheuristic approach for the capacitated arc routing problem. *European Journal of Operational Research*, *253*, 25–39.

Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., & Vogel, U. (2011). Vehicle routing with compartments: applications, modelling and heuristics. *OR Spectrum*, *33*, 885–914.

El Fallahi, A., Prins, C., & Wolfler Calvo, R. (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, *35*, 1725–1741.

Even, S., Itai, A., & Shamir, A. (1976). On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, *5*, 691–703.

Frederickson, G. N. (1979). Approximation algorithms for some postman problems. *Journal of the Association for Computing Machinery*, *26*, 538–554.

Ghiani, G., Laganà, D., Manni, E., Musmanno, R., & Vigo, D. (2014). Operations research in solid waste management: A survey of strategic and tactical issues. *Computers & Operations Research*, *44*, 22–32.

Golden, B. L., & Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, *11*, 305–315.

Henke, T., Speranza, M. G., & Wäscher, G. (2015). The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, *246*, 730–743.

Hertz, A., Laporte, G., & Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. *Operations Research*, *48*, 129–135.

Kiilerich, L., & Wøhlk, S. (2018). New large-scale data instances for CARP and new variations of CARP. *INFOR: Information Systems and Operational Research*, *56*, 1–32.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, *249*, 1–21.

Lacomme, P., Prins, C., & Ramdane-Cherif, W. (2004). Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, *131*, 159–185.

Luiz Usberti, F., França, P. M., & França, A. L. M. (2013). GRASP with evolutionary path-relinking for the capacitated arc routing problem. *Computers & Operations Research*, *40*, 3206–3217.

Mirzaei, S., & Wøhlk, S. (2019). A branch-and-price algorithm for two multi-compartment vehicle routing problems. *EURO Journal on Transportation and Logistics*, *8*, 1–33.

Muyldermans, L., & Pang, G. (2010a). A guided local search procedure for the multi-compartment capacitated arc routing problem. *Computers & Operations Research*, *37*, 1662–1673.

Muyldermans, L., & Pang, G. (2010b). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research*, *206*, 93–103.

Muyldermans, L., & Pang, G. (2014). Variants of the capacitated arc routing problem. In Á. Corberán, & G. Laporte (Eds.), *Arc Routing: Problems, Methods and Applications* chapter 10. (pp. 223–253). Philadelphia: MOS-SIAM Series on Optimization.

Prins, C. (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, *22*, 916–928.

Prins, C. (2014). The capacitated arc routing problem: Heuristics. In Á. Corberán, & G. Laporte (Eds.), *Arc Routing: Problems, Methods, and Applications* chapter 7. (pp. 131–157). Philadelphia: MOS-SIAM Series on Optimization.

Prins, C., Labadi, N., & Reghioui, M. (2009). Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, *47*, 507–535.

Santos, L., Coutinho-Rodrigues, J., & Current, J. R. (2010). An improved ant colony optimization based algorithm for the capacitated arc routing problem. *Transportation Research Part B: Methodological*, *44*, 246–266.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO-Operations Research*, *33*, 1–14.

van der Bruggen, L., Gruson, R., & Salomon, M. (1995). Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research*, *81*, 460–473.

Vidal, T. (2016). Technical note: Split algorithm in *O(n)* for the capacitated vehicle routing problem. *Computers & Operations Research*, *69*, 40–47.

Vidal, T. (2017). Node, edge, arc routing and turn penalties: Multiple problems - one neighborhood extension. *Operations Research*, *65*, 992–1010.

Wang, Q., Ji, Q., & Chiu, C.-H. (2014). Optimal routing for heterogeneous fixed fleets of multicompartment vehicles. *Mathematical Problems in Engineering*, *1*, 1–11.

Wøhlk, S., & Laporte, G. (2018). A fast heuristic for large-scale capacitated arc routing problems. *Journal of the Operational Research Society*, *69*, 1877–1887.

Zbib, H. (2019). *Topics in the Optimization of Waste Collection Systems*. Ph.D. thesis Department of Economics and Business Economics, Aarhus University.

Zbib, H., & Wøhlk, S. (2019). A comparison of the transport requirements of different curbside waste collection systems in Denmark. *Waste Management*, *87*, 21–32.

Appendix A. Detailed results

Table A.1: Detailed computational results for each instance.

Graph	Vehicle	Instance characteristics							Cost	Run time (minutes)				Number of routes	% of vehicles used
		$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{E}_r\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{K}\rvert$	$\lvert\bar{\mathcal{M}}\rvert$	$\sum_{k\in\mathcal{K}} b^k$		Algorithm	Phase 1	Phase 2	Phase 3		
F13_B	M3-2	26	33	19	3	4	3	17	5012	0.004	0.001	0.002	0.001	6	35%
F13_D	M3-1	26	33	19	4	4	4	17	3501	0.01	0.002	0.01	0.001	5	29%
F13_E	M3-1	26	33	19	6	4	4	17	5012	0.5	0.3	0.2	0.005	4	24%
F12_B	M4-2	80	110	72	3	6	3	40	18043	0.1	0.002	0.04	0.1	11	28%
F12_D	M4-1	80	110	72	4	6	4	40	22156	0.8	0.03	0.2	0.5	12	30%
F12_E	M4-1	80	110	72	6	6	4	40	23978	4.0	1.2	1.9	0.7	10	25%
O13_B	M3-2	228	247	170	3	4	3	17	75834	0.2	0.001	0.1	0.01	3	18%
O13_D	M3-1	228	247	170	4	4	4	17	51306	0.8	0.1	0.6	0.02	5	29%
O13_E	M3-1	228	247	170	6	4	4	17	82953	3.7	0.7	2.8	0.2	7	41%
F11_B	M4-2	191	267	174	3	6	3	40	55143	2.4	0.002	0.2	2.2	24	60%
F11_D	M4-1	191	267	174	4	6	4	40	61871	122.1	0.8	1.1	120.0	27	68%
F11_E	M4-1	191	267	174	6	6	4	40	78919	137.9	8.6	8.7	120.0	29	73%
S13_B	M3-2	322	374	176	3	4	3	17	139720	0.2	0.0005	0.1	0.02	2	12%
S13_D	M3-1	322	374	176	4	4	4	17	73480	0.7	0.003	0.6	0.01	2	12%
S13_E	M3-1	322	374	176	6	4	4	17	140450	5.9	0.03	4.5	1.3	4	24%
K13_B	M3-2	394	422	283	3	4	3	17	210000	0.6	0.001	0.5	0.1	7	41%
K13_D	M3-1	394	422	283	4	4	4	17	249796	2.5	0.01	1.6	0.8	4	24%
K13_E	M3-1	394	422	283	6	4	4	17	249378	7.0	0.2	6.6	0.2	4	24%
N13_B	M2-1	454	502	366	3	4	3	16	163416	0.9	0.001	0.7	0.2	6	38%
N13_D	M2-1	454	502	366	4	4	3	16	205292	2.0	0.005	1.7	0.2	8	50%
N13_E	M2-1	454	502	366	6	4	3	16	296319	19.2	0.02	18.5	0.5	10	63%
F10_B	M6-2	415	565	377	3	6	3	80	183235	120.9	0.01	0.8	120.0	51	64%
F10_D	M6-1	415	565	377	4	6	4	80	197972	130.1	1.6	7.1	120.1	55	69%
F10_E	M6-1	415	565	377	6	6	4	80	253297	148.5	10.7	16.1	120.1	63	79%
S12_B	M3-2	755	866	407	3	4	3	17	289333	1.2	0.001	0.9	0.2	6	35%
S12_D	M3-1	755	866	407	4	4	4	17	292671	3.6	0.3	3.1	0.2	9	53%
S12_E	M3-1	755	866	407	6	4	4	17	329860	18.3	0.1	17.8	0.2	6	35%
O12_B	M4-2	761	852	535	3	6	3	40	278915	2.1	0.002	1.7	0.3	8	20%
O12_D	M4-1	761	852	535	4	6	4	40	291030	5.3	0.3	4.2	0.7	12	30%
O12_E	M4-1	761	852	535	6	6	4	40	289190	33.7	6.9	25.8	0.7	11	28%
N12_B	M4-2	930	1040	702	3	6	3	40	368642	3.9	0.001	2.1	1.7	11	28%
N12_D	M4-1	930	1040	702	4	6	4	40	463004	7.9	0.5	6.7	0.6	12	30%
N12_E	M4-1	930	1040	702	6	6	4	40	470756	58.3	5.0	52.0	0.6	13	33%
F1_B	M8-2	812	1124	783	3	6	3	160	795748	123.6	0.003	3.0	120.0	106	66%
F1_D	M8-1	812	1124	783	4	6	4	160	956357	139.2	2.4	14.6	120.1	130	81%
F1_E	M8-1	812	1124	783	6	6	4	160	1102222	152.5	15.2	16.6	120.1	119	74%
K12_B	M2-1	1132	1221	803	3	4	3	16	757737	4.5	0.001	4.1	0.3	11	69%
K12_D	M2-1	1132	1221	803	4	4	3	16	1058062	12.5	0.01	11.0	1.3	16	100%
K12_E	M2-1	1132	1221	803	6	4	3	16	1252638	79.2	0.04	78.3	0.3	10	63%
S11_B	M4-2	1564	1805	961	3	6	3	40	640167	33.1	0.001	4.2	28.8	15	38%
S11_D	M4-1	1564	1805	961	4	6	4	40	560414	20.8	0.3	19.7	0.5	13	33%
S11_E	M4-1	1564	1805	961	6	6	4	40	769908	196.9	8.1	111.5	74.7	17	43%
N11_B	M4-2	2142	2419	1606	3	6	3	40	949328	31.7	0.002	7.8	23.7	22	55%
N11_D	M4-1	2142	2419	1606	4	6	4	40	1099173	32.8	1.5	26.8	3.9	26	65%
N11_E	M4-1	2142	2419	1606	6	6	4	40	1452710	319.8	10.6	187.6	120.6	27	68%
O11_B	M6-2	2822	3281	2132	3	6	3	80	1139040	16.4	0.5	11.8	3.7	36	45%
O11_D	M6-1	2822	3281	2132	4	6	4	80	1369680	164.3	0.2	40.0	121.1	64	80%
O11_E	M6-1	2822	3281	2132	6	6	4	80	1849913	310.9	15.1	167.2	122.6	65	81%
S10_B	M5-2	3404	3921	2221	3	6	3	55	1693181	23.0	0.003	21.2	1.3	25	45%
S10_D	M5-1	3404	3921	2221	4	6	4	55	1876793	166.2	1.7	71.7	90.2	37	67%
S10_E	M5-1	3404	3921	2221	6	6	4	55	2206157	443.6	15.2	299.6	122.7	43	78%
K11_B	M5-2	3114	3361	2281	3	6	3	55	1473116	38.2	0.001	20.9	16.9	32	58%
K11_D	M5-1	3114	3361	2281	4	6	4	55	1756201	194.3	0.6	71.0	120.9	42	76%
K11_E	M5-1	3114	3361	2281	6	6	4	55	1970267	456.7	14.1	315.4	122.7	46	84%
N10_B	M5-2	3698	4187	2802	3	6	3	55	1784650	157.1	0.02	36.4	120.1	34	62%
N10_D	M5-1	3698	4187	2802	4	6	4	55	2129745	195.1	0.5	71.3	121.2	45	82%
N10_E	M5-1	3698	4187	2802	6	6	4	55	2916733	562.3	14.1	422.3	122.7	40	73%
K10_B	M6-2	5102	5518	3744	3	6	3	80	2530588	51.9	0.004	46.9	4.2	44	55%
K10_D	M6-1	5102	5518	3744	4	6	4	80	4894149	255.4	0.5	123.9	123.1	67	84%
K10_E	M6-1	5102	5518	3744	6	6	4	80	3594284	592.2	15.4	442.7	125.6	80	100%
S1_B	M6-2	6149	7110	3797	3	6	3	80	2605964	172.5	0.01	50.6	120.7	47	59%
S1_D	M6-1	6149	7110	3797	4	6	4	80	6040069	256.7	1.9	125.2	123.2	78	98%
S1_E	M6-1	6149	7110	3797	6	6	4	80	4094894	707.9	15.8	559.3	125.2	74	93%

Table A.2: Algorithmic details for each instance.

Graph	Vehicle	Num. of assignments	% of assignments	CSTSA iterations	$\lvert\mathcal{R}\rvert$	Route-vehicle pairs
F13_B	M3-2	22	46%	100	365	1041
F13_D	M3-1	36	11%	140	654	1330
F13_E	M3-1	103	7%	112	1576	3364
F12_B	M4-2	29	35%	100	1980	7932
F12_D	M4-1	76	18%	140	4155	22474
F12_E	M4-1	177	10%	89	6404	37204
O13_B	M3-2	19	40%	100	707	2250
O13_D	M3-1	47	14%	140	2458	10491
O13_E	M3-1	106	7%	79	2929	7996
F11_B	M4-2	45	54%	100	4573	27704
F11_D	M4-1	107	25%	140	11456	72576
F11_E	M4-1	298	17%	91	13842	98716
S13_B	M3-2	16	33%	100	520	1451
S13_D	M3-1	31	9%	140	1052	2648
S13_E	M3-1	75	5%	127	2396	4440
K13_B	M3-2	21	44%	100	808	2137
K13_D	M3-1	39	11%	140	1689	5020
K13_E	M3-1	92	6%	72	2195	4725
N13_B	M2-1	21	44%	100	1134	2862
N13_D	M2-1	36	36%	100	1897	5426
N13_E	M2-1	105	36%	93	3872	14987
F10_B	M6-2	44	52%	100	9530	38638
F10_D	M6-1	112	27%	138	20980	117815
F10_E	M6-1	329	18%	46	15362	107408
S12_B	M3-2	27	56%	100	1431	5050
S12_D	M3-1	47	14%	140	2477	10974
S12_E	M3-1	117	8%	91	3860	9771
O12_B	M4-2	26	31%	100	1651	6756
O12_D	M4-1	69	16%	140	5090	28869
O12_E	M4-1	191	11%	78	7379	34759
N12_B	M4-2	29	35%	100	2275	8065
N12_D	M4-1	69	16%	140	4711	26180
N12_E	M4-1	195	11%	87	7572	37102
F1_B	M8-2	48	57%	100	18583	72452
F1_D	M8-1	133	32%	111	26846	114445
F1_E	M8-1	406	22%	22	7080	45770
K12_B	M2-1	26	54%	100	1595	6888
K12_D	M2-1	44	44%	100	3448	9796
K12_E	M2-1	111	38%	83	5431	19782
S11_B	M4-2	28	33%	100	2297	10215
S11_D	M4-1	76	18%	140	6848	36067
S11_E	M4-1	204	11%	81	8561	43377
N11_B	M4-2	37	44%	100	5253	23155
N11_D	M4-1	102	24%	140	10027	66918
N11_E	M4-1	286	16%	67	13721	92722
O11_B	M6-2	51	61%	100	11554	73336
O11_D	M6-1	101	24%	140	30118	170240
O11_E	M6-1	386	21%	44	24516	211403
S10_B	M5-2	45	54%	100	7666	49847
S10_D	M5-1	113	27%	138	22129	143118
S10_E	M5-1	344	19%	56	22120	165828
K11_B	M5-2	42	50%	100	8682	47356
K11_D	M5-1	109	26%	136	23738	146413
K11_E	M5-1	363	20%	58	24969	189053
N10_B	M5-2	36	43%	100	8141	34539
N10_D	M5-1	105	25%	140	20312	132115
N10_E	M5-1	335	19%	50	20569	170950
K10_B	M6-2	47	56%	100	13678	91001
K10_D	M6-1	112	27%	112	34802	225016
K10_E	M6-1	381	21%	40	25026	220165
S1_B	M6-2	50	60%	100	15434	106637
S1_D	M6-1	131	31%	122	36634	299688
S1_E	M6-1	387	21%	40	26090	241563

Credit Author Statement

Hani Zbib: conceptualization; formal analysis; methodology; software; validation; data processing; writing.

Gilbert Laporte: counseling and supervision; funding; writing.