A Districting-Based Heuristic for the Coordinated Capacitated Arc Routing Problem

Sanne Wøhlk[a,*], Gilbert Laporte[b]

[a] *CORAL - Cluster for Operations Research, Analytics, and Logistics, Department of Economics and Business Economics, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark.*

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

Abstract

The purpose of this paper is to solve a multi-period garbage collection problem involving several garbage types called *fractions*, such as general and organic waste, paper and carboard, glass and metal, and plastic. The study is motivated by a real-life problem arising in Denmark. Because of the nature of the fractions, not all of them have the same collection frequency. Currently the collection days for the various fractions are uncoordinated. An interesting question is to determine the added cost in terms of traveled distance and vehicle fleet size of coordinating these collections in order to reduce the inconvenience borne by the citizens. To this end we develop a multi-phase heuristic: 1) small collection districts, each corresponding to a day of the week, are first created; 2) the districts are assigned to specific weekdays based on a closeness criterion; 3) they are balanced in order to make a more efficient use of the vehicles; 4) collection routes are then created for each district and each waste fraction by means of the FastCARP heuristic. Extensive tests over a variety of scenarios indicate that coordinating the collections yields a routing cost increase of 12.4%, while the number of vehicles increases in less than half of the instances.

Keywords: Garbage collection, districting, arc routing, heuristics

---

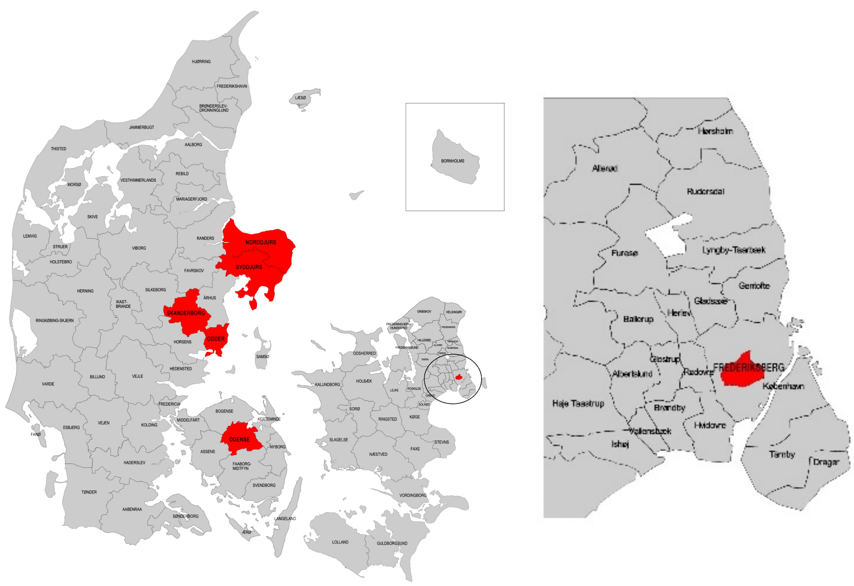
1. Introduction

The purpose of this paper is to solve a multi-period garbage collection problem involving several garbage types called *fractions*. In our application these are

*Corresponding author

Email addresses: sanw@econ.au.dk (Sanne Wøhlk), gilbert.laporte@cirrelt.ca (Gilbert Laporte)

general and organic waste, paper and carboard, glass and metal, and plastic. Our study is motivated by a real-life problem arising in Denmark. In this country, garbage collection falls under the responsibility of the counties. There are 98 Danish counties and we obtained data for six of them under a cooperative research agreement (see Figure 1). These counties represent several areas of Denmark. Two are rural (North (N) and South (S) Djurs), two are semi-rural (Skanderborg and Odder (K)) and will be treated as one in our experiments, and two are urban (Frederiksberg (F) and Odense (O)). Because of the nature of the fractions (for example, organic waste attracts animals), not all of them require the same collection frequency. While general or organic garbage may require weekly or biweekly collections, recyclable materials such as glass and paper may have longer collection intervals. Under the current practice, the collection days for different fractions are uncoordinated. To illustrate, general and organic waste can be collected every Monday, paper and cardboard every fourth Tuesday, glass and metal every third Thursday, and plastic every second Friday (see Figure 2).



Denmark. *Zoom on Copenhagen.*

Figure 1: The counties providing the data.

However, in places like Denmark, where the citizens have to move their waste bins from the backyard to the sidewalk the evening before collection, and back the next day, such a schedule means that the citizens must relocate bins multiple days each week. It would be easier for them to perform such operations all at once, and they would not have to be constantly preoccupied with garbage collection. For this reason, a more coordinated collection schedule, such as the one depicted in Figure 3, could be preferable. In this schedule, garbage collection always takes place on a Monday for all fractions, even though the garbage type varies from week to week. The idea of introducing coordinated collection schedules for garbage collection, where all collections would always occur on the same weekday for each citizen, was recently proposed in [1]. Clearly, a coordinated schedule may not be possible in some old towns with narrow streets where there is not sufficient space to put many bins all at once. However, in the majority of the places underlying our study in Denmark, there are no real practical problems with a coordinated schedule.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
General and organic	X	X	X	X	X	X	X	X	X	X	X	X
Paper and cardboard	X				X				X			
Glass and metal			X			X			X			X
Plastic		X		X		X		X		X		X

Figure 2: Example of an inconvenient schedule for the citizens.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
General and organic	X	X	X	X	X	X	X	X	X	X	X	X
Paper and cardboard	X				X				X			
Glass and metal			X			X			X			X
Plastic		X		X		X		X		X		X

Figure 3: Example of a more convenient schedule for the citizens.

There is yet no hard computational evidence that the uncoordinated collection schedules are beneficial in terms of collection costs. Some of our partners have expressed their interest in an investigation of the relative collection cost of the coordinated and uncoordinated schedule. The research question that motivates this paper is therefore to determine by how much implementing a coordinated collection schedule would increase the collection costs, measured as the number of vehicles needed (assuming that a vehicle can perform a single route each day), as well as the total distance driven. In order to provide an input to the decision process, we have developed a districting-based heuristic for the design of coordinated garbage collection schedule, and we have compared the solutions to those obtained under an uncoordinated collection scheme. Our computational results indicate that the increase in distance is 12.4% and the number of vehicles increases in less than half of the cases. Whether such increases are acceptable or not is up to the administrators to decide.

The problem under consideration is of very large scale, which imposes severe practical restrictions on our solution methodology. To give the reader an idea of the large size of these instances, in the five areas considered in this study, there are between 26 and 11,656 nodes, between 33 and 12,691 edges, between 19 and 8,651 required edges, and between two and 54 vehicles over all fractions. These very large sizes preclude the use of any exact algorithm (for example, the largest CARP instances that can be solved optimally involve around 190 edges (see [2])). The use of metaheuristics such as tabu search, adaptive large neighbourhood search, or genetic algorithms (see [3] for a survey) is also impractical since such techniques must apply several destroy and repair operators over a very large number of iterations in order to produce high quality results. Typically, if a problem instance involves n nodes, then the complexity of most standard operators tends to be at least $O(n^2)$. As a result, we must contend ourselves with relatively simple low-complexity heuristics. It soon became clear to us that as a first step we would have to partition the counties into more manageable districts in each of which a routing heuristic would have to be applied separately. One such heuristic, used as a subroutine in this work, is the FastCARP algorithm recently proposed by [4] for the CARP. As we will show, the use of districts not only reflects the current managerial practice, but it is instrumental in the design of coordinated collection schedules. However, the districting process is not without some inherent difficulties, as we will explain in the following.

As far as we are aware, the problem under study has never been previously studied. Related papers are those of [5], [6], and [7] for territorial districting in an arc routing context, [8], [9], and [10] for the study of multi-fractions garbage collection problems, [3] for the design of heuristics for the CARP in general, and [11] and [12] for the CARP in a garbage collection context. The papers by [13] and [14] also describe interesting case studies in the context of curbside garbage collection, while [15] takes a broader perspective on research of garbage collection operations. However, it looks as if the design of coordinated multi-fractions collection schedules has never been investigated. Furthermore, we refer the reader to [16] and [17] for thorough surveys of the CARP.

The remainder of this paper is organized as follows. Section 2 contains a detailed description of our problem. In Section 3, we discuss different approaches to decomposing the problem into districts and explain why our approach to districting makes sense not only in terms of working with smaller and more manageable territorial entities, but also as a means of coordinating the collection schedules. Our solution approach is described in Section 4 and Section 5 describes our base of comparison without enforcing coordination. Extensive computational experiments are presented in Section 6, followed by conclusions in Section 7.

2. Formal Problem Description

The problem just introduced is known as the Coordinated Capacitated Arc Routing Problem (C-CARP) and was first presented in [1]. In the following, we will describe the problem formally, using a slightly simplified notation.

The C-CARP is defined on an undirected connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The edges are defined as unordered pairs (i, j) with $i, j \in \mathcal{N}$, and with every edge, $(i, j) \in \mathcal{E}$ is associated a traversal cost $c_{ij} > 0$. With every pair of nodes i and j, we associate a cost s_{ij} equal to the shortest distance between i and j, which can easily be determined. Node $0 \in \mathcal{N}$ represents the depot.

In order to describe the time perspective of the problem, let τ be the number of service days in a week, i.e. τ is typically 5 or 6. Let $\mathcal{T} = \{1, 2, 3, \ldots\}$ be the set of service days in the planning horizon. Note that $\mathcal{T}$ excludes non-service days, and therefore, if $\tau = 5$, then days $1, 6, 11, \ldots$ are Mondays.

We denote by $\mathcal{F}$ the set of all waste fractions to be collected, and for each fraction $f \in \mathcal{F}$, a demand $q^f_{ij} \geq 0$ must be collected from edge (i, j) every l^f days with respect to $\mathcal{T}$ (e.g. if $\tau = 6$, then $l^f = 6$ and $l^f = 12$, correspond to weekly and biweekly collection, respectively). We define $\mathcal{E}^f_R = \{(i, j) \in \mathcal{E} : q^f_{ij} > 0\}$. By definition, we have either $(l^f \mod \tau) = 0$ or $(\tau \mod l^f) = 0$ for every fraction $f \in \mathcal{F}$, i.e., $l^f \in \{\ldots, \tau/3, \tau/2, \tau, 2\tau, 3\tau, \ldots\}$. We refer to waste fractions with $l^f < \tau$ as frequent fractions and to those with $l^f \geq \tau$ as non-frequent fractions.

To collect the waste, a set $\mathcal{K}^f$ of identical vehicles are available for each waste fraction $f \in \mathcal{F}$. Each vehicle for collection of fraction f has a capacity equal to W^f.

A feasible solution of the C-CARP is characterized by sets of routes for the vehicles satisfying the following requirements:

1. every waste fraction $f \in \mathcal{F}$ is collected from every edge (i, j) with $q^f_{ij} > 0$ every l^f days throughout the planning horizon;
2. for each edge, the collection of all non-frequent waste fraction is done on the same day of the week;
3. for each edge, one of the weekly collections of each frequent fraction is done on the same day of the week as the collection of the non-frequent fractions on that edge;
4. each vehicle performs at most one route each day in $\mathcal{T}$;
5. waste fraction $f \in \mathcal{F}$ is collected only by vehicles in $\mathcal{K}^f$;
6. the total demand collected by each vehicle each day does not exceed the capacity of that vehicle;
7. all routes start and end at the depot;
8. every collection of a given waste fraction from a given edge is done by the same vehicle.

The primary objective of the C-CARP is to minimize the total number of vehicles used, while the secondary objective is the minimization of the total routing cost over the $|\mathcal{T}|$ days of the planning horizon, defined as the total distance traversed by the vehicles.

3. Decomposition of the problem

When solving very large scale and complex problems such as the one considered in this paper, it is highly sensible to apply some kind of decomposition. The most natural approach would be to decompose the problem by waste fraction such that each waste fraction is treated separately. However, this is not possible due to the coordination constraints, and this approach was therefore discarded. An alternative approach is to decompose the problem graphically. While this would decrease the size of each subproblem, it does not reduce the complexity of the problem.

Finally, the problem can be decomposed along the time horizon and through the use of districts. A district is defined as a set of edges serviced on a specific day of the week. There are two ways this decomposition can be implemented. First, the problem can be decomposed by calendar day, so that each day in $\mathcal{T}$ is handled separately. To this end, we can create the planning horizon $|\mathcal{T}|$ districts, one for each day, and assign each waste fraction of each edge to several districts that are l^f days apart in time and in such a way that the demand for each fraction is balanced across the districts. This would result in 12 to 60 districts, depending on the instance. While performing this assignment and possibly adjusting the assignment, the coordination requirements 2 and 3 is Section 2 must be respected. Once the districts are designed, routes must be created to service the edges of each district. During the routing phase, requirement 8 must be ensured, possibly by replication of the same set of routes for several districts.

Second, the problem can be decomposed on the basis of *weekdays*, as opposed to specific days of the planning horizon, by forming one district for Mondays, one for Tuesdays, and so forth. This approach results in τ districts, each edge in the graph being assigned to exactly one district, thereby automatically enforcing the coordination requirements 2 and 3. When seeking to obtain well-balanced districts, the demand of frequent edges needs extra care, as illustrated in the next paragraph, but non-frequent edges are handled in a straightforward manner. After creating routes for each district and each fraction, a finalization procedure needs to be performed. To illustrate this, consider the Monday district for a fraction with $l^f = 3\tau$, and assume that 15 routes are created for it. Now, the first five routes are executed on Monday in the first week, the next five on Monday the second week, and the final five on Monday the third week, and this plan is repeated the following Mondays in a cyclic manner. This is the approach we have chosen for decomposing the problem in this paper because the coordination can be handled more naturally and the balancing of districts is easier due to the smaller number.

Under this approach, the handling of non-frequent fractions is straightforward, but extra attention still needs to be given to frequent fractions when balancing the districts. This is illustrated in the following example with $\tau = 6$, and three fractions with $l^1 = 6, l^2 = 3$, and $l^3 = 2$. Here, six districts are created and each edge is assigned to one district. The districts are assigned to weekdays: district

		Weekdays					
		1	2	3	4	5	6
Fraction	1	**1**	**2**	**3**	**4**	**5**	**6**
	2	**1** 4	**2** 5	**3** 6	1 **4**	2 **5**	3 **6**
	3	**1** 3 5	**2** 4 6	1 **3** 5	2 **4** 6	1 3 **5**	2 4 **6**

Table 1: Districts associated with each waste fraction and each weekday in the example of Section 3.

1 is assigned to Mondays, district 2 to Tuesdays, and so forth. In Table 1 we show, for each waste fraction, which district is being collected each weekday. In the figure, we have marked in bold the district that is assigned to the weekday. For instance, we see that on weekday 3, we collect waste fraction 3 from the edges assigned to districts 1, 3, and 5. In our algorithm, we ensure that the same routes are used for collecting fraction 3 on weekdays 1, 3, 5, when these districts are serviced.

4. Description of the algorithm

Our overall solution strategy for the C-CARP is outlined in Algorithm 1 and detailed in the following sections. We start with a short overview.

Algorithm 1 Overview of the full algorithm

Create initial districts ▷ Section 4.1
Assign districts to weekdays ▷ Section 4.2
if the districts are unbalanced **then**
  Balance districts using Algorithm 3 ▷ Section 4.3
end if
for each $f \in \mathcal{F}, d \in \{1, \dots, \min\{\tau, l^f\}\}$ **do**
  Call FastCARP(f, d) ▷ Section 4.4
end for
Finalize solution ▷ Section 4.5

We start by partitioning $\mathcal{G}$ into τ initial districts in such a way that every edge is assigned to exactly one district. Because the demands of the different waste fractions across the edges are not perfectly correlated, and because the frequent fractions need to be collected on multiple weekdays, the algorithm does not attempt to generate perfectly balanced districts at this point. Each district will subsequently be assigned to a day of the week d such that all waste fractions of the edges in the district are collected on day d (frequent fractions are also collected at one or more additional days). Thereby, the districts will ensure the coordination, which is the purpose of their creation. Applying this procedure ensures that requirement 1 of Section 2 is satisfied and that the coordination requirements 2 and 3 are also dealt with. Creation of the initial districts is detailed in Section 4.1.

After the creation of the initial weekday districts, we assign them to specific weekdays based on a combination of closeness of the districts and of the interaction among the days. This is detailed in Section 4.2.

Based on the knowledge of collection intervals for each waste fraction and the edges assigned to the district, we know the total amount to be collected on each day, and hence we can estimate the number of vehicles needed to collect that waste. For non-frequent fractions, this is relatively straightforward. But for frequent fractions, we must take into consideration that for instance, for a fraction with $l^f = 3$ ($\tau = 6$), both the edges assigned to the district of day 1 and those assigned to the district of day 4 must be serviced on both days. Next, we calculate a slack value for each waste fraction and each district based on the initial districts and their assignment to days of the week. Intuitively, this tells us whether a district requires more vehicles than predicted by a lower bound (if the slack is positive) or fewer vehicles (if the slack is negative). If any of the districts requires too many vehicles, we seek to obtain a better balancing among the districts. The main challenges in the balancing phase comes from the fact that we work with multiple waste fractions simultaneously, not all of which require collection from all edges, as well as from the frequent fractions because they affect multiple districts when moved. The purpose of this balancing phase is to favour the primary objective of the problem: minimizing the total number of vehicles needed. The balancing phase preserves the coordination and is described in Section 4.3.

After the districts have been determined and assigned to days of the week, the problem reduces to a CARP for each day of the week and each waste fraction, thus satisfying requirements 5, 6, and 7. This problem is solved by means of the FastCARP heuristic developed in [4], which is summarized in Section 4.4. It is also here that we handle requirement 8. This heuristic aims to favour the secondary objective: minimizing the total routing cost.

The solution is then finalized by determining which routes to service each week while satisfying requirement 4, and the final total cost calculation is performed. This is detailed in Section 4.5. During this process, we also ensure that the waste of an edge is always collected by the same vehicles, one for each fraction.

4.1. Creation of the Initial Districts

Our districting algorithm consists of two phases. We first create a number of small districts, and then merge them into weekday districts.

In the first phase, we first determine the minimum number $\hat{\Phi}$ of small districts that we aim for. Based on preliminary tuning, we set $\hat{\Phi} = 2\tau$. However, due to variations in demand, we tend to end up with significantly more than $\hat{\Phi}$ small districts, in particular when the vehicles are relatively small compared with the demand, or when the demand for different fractions is unevenly spread over the graph. For each fraction $f \in \mathcal{F}$, we define $L^f = (\sum_{(i,j)\in\mathcal{E}_R^f} q_{ij}^f)/\hat{\Phi}$ as the amount of demand of each fraction we aim for in each small district. To create a small district, we start by identifying a seed, which is chosen as the node furthest from the depot having at least one unassigned adjacent edge. We then repeatedly select an edge with one end node closest to the seed, and secondarily, furthest from the depot as a candidate to be added to the district. If the candidate edge does not cause the total sum of demand to exceed L^f for any $f \in \mathcal{F}$, it is added to the district. This process is iterated until no more edges can be added without exceeding L^f for some fraction $f \in \mathcal{F}$. We repeat the entire process until all edges have been assigned to a district. After the first phase, we have obtained Φ small districts, usually with $\Phi > \hat{\Phi}$. At this point, we redefine the seeds of the districts by identifying the node adjacent to a required edge in the district, for which the sum to all other endpoints of required edges in the district is minimized, and we designate that node as the seed of the district. With this definition, the seed represents a "centre of gravity" of the required edges in the district.

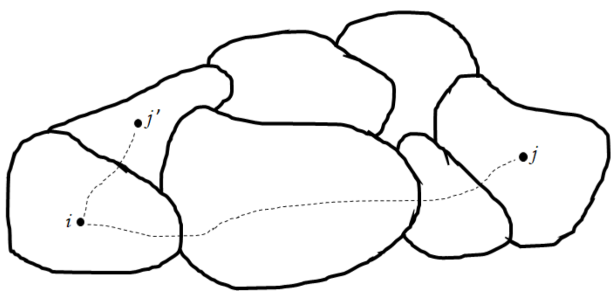


Figure 4: Illustration of the modified distance function. Based on that, $\text{dist}(i, j) = 4$ and $\text{dist}(i, j') = 2$.

In the second phase, we make use of a modified distance function, which defines the distance between any two nodes as the number of districts that the shortest path between the nodes intersects. This is illustrated in Figure 4. In this phase, we merge the Φ small districts into τ weekday districts by repeatedly selecting a father district and a non-father district to merge as described below, letting the seed of the joint district be the new centre of gravity of the joint district. For this process, we define $\overline{L}^f$ similarly to L^f, but now with the goal of creating τ weekday districts, hence $\overline{L}^f = (\sum_{(i,j) \in \mathcal{E}_R^f} q_{ij}^f)/\tau$. We also define a buffer ρ which, based on preliminary tuning, is initially set to 1.2. We first identify a set of τ father districts, one for each service day of the week, as follows. The first father district is selected as the one that maximizes the distance between the depot and the seed of the district. The remaining $\tau - 1$ father districts are selected iteratively as the district whose seed node is furthest away from the closest seed node of the existing father districts. In case of a tie, we use the modified distance function as secondary criterion. We now repeatedly consider all father - non-father pairs, and among the pairs we select one for which the joint demand does not exceed $\rho\overline{L}^f$ for any fraction f. The primary selection criterion is the minimization of the modified distance function, and the secondary criterion is the minimization of the shortest distance between the seeds of the districts. We merge the non-father district with the father district and update the seed to represent the centre of gravity of the joint district. This process is repeated while pairs are found within the limit of $\rho\overline{L}^f$. When no further pair can be identified, ρ is multiplied by 1.2, and the search process is reiterated. The full process if repeated until only τ districts remain and these are the initial weekday districts. The logic behind this 2-phase process is that the demand is very unevenly distributed over the graph based on distances, but that it is more evenly distributed over the small districts and using the modified distance function. This eases the creation of relatively balanced districts. Figure 5 provides an example of the initial districts in one of our instances.

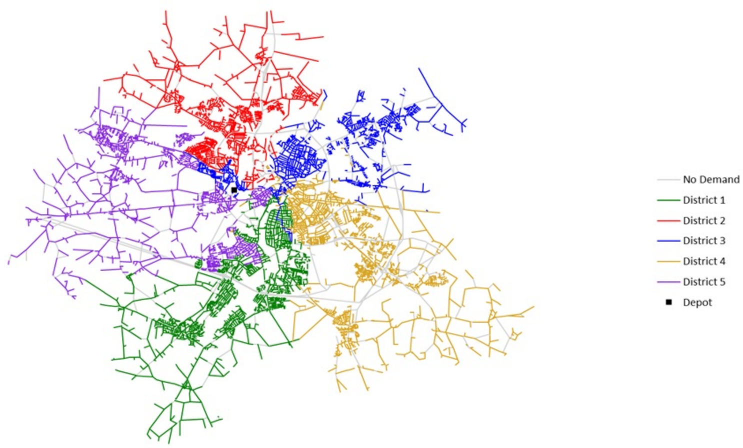


Figure 5: Example of initial districts. Here showing instance O1_A.

4.2. Assigning the Districts to Weekdays

When $l^f \geq \tau$ for all waste fractions f, it does not matter how the districts are assigned to weekdays since there will be no interaction between the districts. Hence we perform the assignment in a straightforward manner. However, this is not the case when frequent fractions are present. Consider, for example a fraction requiring service twice a week, with $\tau = 6$. In this example, the districts assigned to days 1 and 4 will both be serviced for this fraction on both days (as well as districts assigned to days 2 and 5, for instance). In Figure 6, we show two different assignments of districts to weekdays. In the straightforward assignment shown in the lower part of the figure, the districts to be collected jointly are geographically distant, whereas in the upper assignment the joint collection occurs from neighbouring districts. To foster good routing, our procedure aims to create assignments with the characteristics of the upper assignment. Viewed

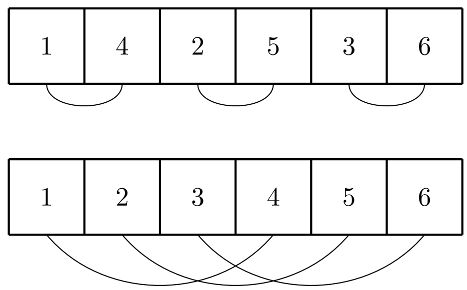


Figure 6: Illustration of two different assignments of six districts to weekdays, where each square represent a district. The numbers in the districts represent weekdays and the arcs represent districts to be collected jointly for a waste fraction with $l^f = 3$.

in a different way: the more interaction between two weekdays in terms of coinciding collections, the more important is it that the districts assigned to those days are not too distant from each other. This is the motivation behind the following procedure for the assignment of districts to weekdays when at least one waste fraction f has $l^f < \tau$.

To measure the distance between two districts, we use the shortest path distance s_{ij} between the seeds of the districts, as representatives of the centre of gravities. To measure the interrelation between the weekdays, we construct a matrix A of size $\tau \times \tau$. For each pair of days, $i, j \in \{1, \dots, \tau\}$, the value of A_{ij} is the number of days during the week for which the districts assigned to i and j, respectively, will be collected jointly. The values of the A matrix are determined by Algorithm 2.

Algorithm 2 Construction of the A matrix

$A_{ij} = 0 \ \forall i, j \in \{1, \dots, \tau\}$
for all f frequent **do**
  for all $i, j \in \{1, \dots, \tau\}, i \neq j$ **do**
    if $(i \mod l^f) = (j \mod l^f)$ **then** $A_{ij} \leftarrow A_{ij} + \tau / l^f$
    end if
  end for
end for

To illustrate this algorithm, consider an example with $\tau = 6$ and two frequent fractions: fraction 1 with collection interval $l^1 = 3$ (two collections per week), and fraction 2 with collection interval $l^2 = 2$ (three collections per week). Table 2 shows the A matrix for this example. Here, $A_{14} = 2$ means that twice during the week, the districts assigned to weekdays 1 and 4 must be collected together (this will happen on weekdays 1 and 4); $A_{13} = 3$ means that three times per week, the districts assigned to days 1 and 3 must be collected together (this will happen on weekdays 1, 3, and 5). Therefore, the higher the value A_{ij}, the more costly it is for the districts assigned to weekdays i and j to be far from each other.

	1	2	3	4	5	6
1	0	0	3	2	3	0
2	0	0	0	3	2	3
3	3	0	0	0	3	2
4	2	3	0	0	0	3
5	3	2	3	0	0	0
6	0	3	2	3	0	0

Table 2: Illustration of matrix A.

Let $\mathcal{O}$ be the set of all possible vectors O of length τ representing assignments of district seeds to weekdays i ($i = 1, \ldots, \tau$). This definition of O induces a one-to-one correspondence between weekdays and districts. We seek the least costly assignment in terms of joint collection from multiple districts on the same day, and thereby we select the assignment

$$O^* = \arg\min_{O \in \mathcal{O}} \{\sum_{i=1}^{\tau} \sum_{j=1}^{\tau} s_{O(i)O(j)} A_{ij}\}.$$

The number of distinct assignments is $(\tau - 1)!$ which is not very large since τ is the number of service days per week (120 if $\tau = 6$). We therefore determine O^* by full enumeration.

Finally, and independently of the method used for assigning districts to weekdays, we renumber the districts in such a way that the district assigned to day i is indexed by i.

4.3. Balancing of the Districts

To motivate the next part of the algorithm, we consider an example, where, for some waste fraction, two vehicles are needed to collect the waste of edges of the district assigned to Mondays, while four are needed on Tuesdays, and six are needed on Wednesdays. To collect the waste in this manner, six vehicles are necessary. A more balanced assignment of edges to districts would result in only four vehicles being needed in this example. When multiple waste fractions are available, obtaining balanced districts is more involved than with a single fraction because the same district may need too many vehicles for one fraction, while at the same time needing too few vehicles for another fraction. As an additional complication, we face the fact that for frequent fractions, moving an edge from one district to another not only affects the waste to be collected in these two districts, but also affects the districts assigned to the other days when this edge must be serviced. In this section, we describe our algorithm to move edges between districts in order to obtain a more balanced partitioning of the graph into districts, and thereby favour the primary objective of the problem.

To describe the balancing procedure, we need some additional notation. With the renumbering at the end of Section 4.2, we know that district number d is assigned to weekday d. We use $\mathcal{E}_d$ to denote the set of edges assigned to district d. For each frequent fraction f and each weekday d, we define P_d^f as the set of weekdays that are multiples of l^f days away from d, i.e. $P_d^f = \{d' \in \{1, \ldots, \tau\} : d' = d + \alpha l^f, \alpha \text{ integer}, \alpha \neq 0\}$. Hence, on day d, when collecting fraction $f \in \mathcal{F}$, we must collect from the district assigned to day d as well as from the districts assigned to the days in P_d^f.

For every waste fraction $f \in \mathcal{F}$ and every weekday d, we compute a lower bound $\hat{K}_d^f$ on the number of vehicles needed to service the demand of fraction f assigned to that day. For non-frequent waste fractions, this bound is computed as $\hat{K}_d^f = \lceil \frac{\tau}{l^f W^f} \sum_{(i,j) \in \mathcal{E}_d} q_{ij}^f \rceil$. For frequent fractions, both the demand of the district assigned to day d and the demand in districts assigned to days that are a multiple of l^f away from d must be serviced on day d. Hence, the number of vehicles needed is at least $\hat{K}_d^f = \lceil \frac{1}{W^f} (\sum_{(i,j) \in \mathcal{E}_d} q_{ij}^f + \sum_{d' \in P_d} \sum_{(i,j) \in \mathcal{E}_{d'}} q_{ij}^f) \rceil$.

We also compute the overall minimum number of vehicles needed for each fraction if the demand is evenly spread over all days as $\overline{K}^f = \lceil \frac{\tau}{l^f W^f} \sum_{(i,j) \in \mathcal{E}_R^f} q_{ij}^f \rceil$ for both frequent and non-frequent fractions.

Next, we define the slack S_d^f of each fraction $f \in \mathcal{F}$ and each weekday d as $S_d^f = \hat{K}_d^f - \overline{K}^f$. Intuitively, the slack is the additional number of vehicles needed to service the demand for f on weekday d, compared to the theoretical lower bound on the number of vehicles needed. Therefore, $S_d^f > 0$ is an indication that some demand for fraction f needs to be removed from district d (or from a district in P_d^f if f is frequent) in order to obtain a balanced distribution, whereas $S_d^f < 0$ means that district d (or $d' \in P_d^f$ if f is frequent) can safely receive some additional demand for f. We also define $\overline{S_d} = \max_{f \in \mathcal{F}} \{S_d^f\}$ and $\underline{S_d} = \min_{f \in \mathcal{F}} \{S_d^f\}$, as well as $\overline{S} = \max_{d \in D} \{\overline{S_d}\}$ and $\underline{S} = \min_{d \in D} \{\underline{S_d}\}$, where a perfect balancing will result in these values differing by no more than one.

The idea behind our balancing algorithm as outlined in Algorithm 3, is to decrease $\overline{S}$ as long as possible, then increase $\underline{S}$, and repeat this exchange mechanism as long as changes are found for either to the two values. During the course of the algorithm, we keep track of the boundary B_d of the districts, which we define as the set of nodes adjacent to at least one edge in the district, but also to at least one edge in another district. Whenever an edge is moved from a district d' to another district d'', both $B_{d'}$ and $B_{d''}$ may need to be updated regarding the two end nodes of the edge.

We move demand in two different ways in the algorithm, corresponding to the two 'while blocks', both of which will be detailed below. We first consider the situation where we seek to move edges away from a given weekday d'. In this case, d' is selected to be the district of a weekday with the largest slack $\overline{S}$. We seek to move edges that are adjacent to the boundary $B_{d'}$ to neighbouring districts until the largest slack $\overline{S_{d'}}$ of d' is decreased by one unit (this is controlled

Algorithm 3 Balance districts

Compute $\overline{S_d}$ and $\underline{S_d}$ $\forall d$, and $\overline{S}$ and $\underline{S}$. ▷ Keep them updated
Set improved = true
while improved = true **do** ▷ Repeat while improvement is obtained
    Set improved = false
    Set $U = \{1, \ldots, \tau\}$
    while $U \neq \emptyset$ and $\overline{S} > 0$ **do**
        Set $d' = \arg\max_{d \in \{1,\ldots,\tau\}} \{\overline{S_d}\}$ ▷ Most positive slack
        Set $m = \overline{S_{d'}}$
        while Possible and $\overline{S_{d'}} = m$ **do**
            Move edges from d' to neighbouring districts
        end while
        if $\overline{S_{d'}} = m - 1$ **then** ▷ Decrease slack of weekday d'
            improved = true
        else ▷ No improvement was obtained for d'
            $U = U \setminus \{d'\}$
        end if
    end while
    Set $U = \{1, \ldots, \tau\}$
    while $U \neq \emptyset$ and $\underline{S} < 0$ **do**
        Set $d' = \arg\min_{d \in \{1,\ldots,\tau\}} \{\underline{S_d}\}$ ▷ Most negative slack
        Set $m = \underline{S_{d'}}$
        while Possible and $\underline{S_{d'}} = m$ **do**
            Move edges from neighbouring districts to d'
        end while
        if $\underline{S_{d'}} = m + 1$ **then** ▷ Increase slack of weekday d'
            improved = true
        else ▷ No improvement was obtained for d'
            $U = U \setminus \{d'\}$
        end if
    end while
end while

by the variable m in the algorithm). This is done in the following way. The nodes i in $B_{d'}$ are considered in decreasing order of to their distance to the seed of district d', and the edges adjacent to i which are in district d' are then considered in arbitrary order. For each such edge (i, j), we consider the other weekday districts d that also have i on their boundary. Among those, we move (i, j) to the district where the seed of d is closest to (i, j), and where S_d^f will not be increased for any fraction f by the addition of (i, j) to d. This movement may cause i or j to be added to or removed from the boundaries of d' or d. It may happen that no district can receive (i, j) under the given restrictions, in which case we proceed without moving the edge. The process stops when the largest slack of d' is decreased by one unit or when all edges adjacent to the boundary have been considered. At this point, if $\overline{S_{d'}}$ has not decreased, we temporarily exclude d' from consideration (this is controlled by the set U in the algorithm). The whole process is then repeated until no further improvement in the slack can be found and all districts have been temporarily excluded.

When no further improvement can be found by moving edges *from* specific districts, we start to seek improvements by moving edges *to* specific districts with a negative slack. To this end, we select d' among the weekdays with the most negative slack $\underline{S_{d'}}$, and we seek to increase the slack by one unit (this is again controlled by m in the algorithm). Now, the nodes i in $B_{d'}$ are considered in increasing order of their distance to the seed of district d', and the edges adjacent to i that are not in the district d' are then considered in arbitrary order. The edge is moved from its current district d to district d' provided that the movement does not cause $S_{d'}^f > 0$ for any waste fraction in the receiving district and does not cause the slack $\underline{S_d}$ to become smaller than the current worst slack $\underline{S}$ (note that because we move edges away from d, we risk decreasing the slack of d). Again, the process continues until $\underline{S_{d'}}$ is improved or until all edges adjacent to the boundary of d' have been explored, and the process is reiterated as above. This alternating process is repeated as long as improvements are obtained in either of the two parts.

In Table 3, we illustrate the effect of our balancing procedure on an example. The top of the table shows the least number of vehicles, $\hat{K}_d^f$ needed to collect each waste fraction in each district before (left) and after (right) balancing, as well as the global lower bound on the number of vehicles for each fraction, $\overline{K}^f$. The lower part of the table shows the slack S_d^f for each district and each waste fraction, as well as the upper and lower bounds for each district, before (left) and after (right) balancing. In the left part of the table, we see that $\overline{S} = \max_{d \in D}\{\overline{S_d}\} = 3$, which is obtained for districts 1 and 5. We arbitrarily select one of them: 1. The algorithm thus starts by attempting to move demand from district 1 to the other districts. After moving sufficient demand away from district 1, the slack of that district decreases to $\overline{S_1} = 2$, but we still have $\overline{S_5} = 3$, and thereby $\overline{S} = 3$. The algorithm now attempts to move demand from district 5 to the other districts until $\overline{S_5}$ decreases to 2, at which point $\overline{S} = 2$. This value is again obtained for both districts 1 and 5, and the algorithm will now

		District					District					
		1	2	3	4	5	1	2	3	4	5	
		$\hat{K}_d^f$ before balancing					$\hat{K}_d^f$ after balancing					$\overline{K}^f$
Fraction	1	6	4	5	4	6	5	5	5	5	5	5
	2	17	11	15	12	17	15	14	14	14	14	14
	3	5	3	5	4	5	4	4	5	4	4	4
		S_d^f before balancing					S_d^f after balancing					
Fraction	1	1	-1	0	-1	1	0	0	0	0	0	
	2	3	-3	1	-2	3	1	0	0	0	0	
	3	1	-1	1	0	1	0	0	1	0	0	
$\overline{S_d}$		3	-1	1	0	3	1	0	1	0	0	
$\underline{S_d}$		1	-3	0	-2	1	0	0	0	0	0	

Table 3: Example of balancing of instance O10_B.

proceed by first moving demand from one of these, then from the other, and so on. The right-hand side of Table 3 shows that the districts after completion of the balancing algorithm are better balanced. In fact, the estimated need for vehicles after balancing is 25 (5+15+5), compared to 28 (6+17+5) before the balancing procedure. Figure 7 shows the districts in our example from Figure 5 after the balancing process.

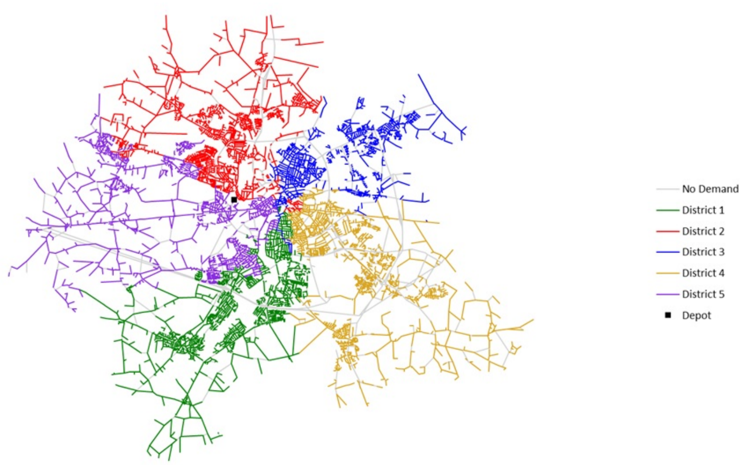


Figure 7: Example of balanced districts. Here showing the same instance as in Figure 5.

4.4. Creation of Routes

At this point in the algorithm, we know which edges to service on each weekday and the route creation can start. For each non-frequent fraction $f \in \mathcal{F}$, the edges in $\mathcal{E}_d^f$ must be serviced on weekday d (e.g. on Mondays for $d = 1$). We use the FastCARP algorithm of [4], summarized below, to create a set $\mathcal{R}_d^f$ of routes servicing these edges. For frequent fractions $f \in \mathcal{F}$, we know that exactly the same set of edges must be serviced on day d as on days $d' \in P_d$, namely the edges in $\mathcal{E}_d \cup \bigcup_{d' \in P_d} \mathcal{E}_{d'}$. We therefore use FastCARP to create a set $\mathcal{R}_d^f$ of routes servicing all of these edges and we repeat the same set of routes for weekdays $d' \in P_d$. Therefore, the route generation procedure is only executed for the first l^f weekdays for frequent fractions. With this procedure, we ensure that the edges in $\mathcal{E}_d$ and $\bigcup_{d' \in P_d} \mathcal{E}_{d'}$ are serviced by the same vehicle every time they are serviced, even though they are assigned to different districts.

The FastCARP algorithm described in [4], which we use to create the routes is designed to solve large-scale CARPs within a short computation time. It starts by creating a giant tour without consideration of vehicle capacities. When considering large-scale problems, the approach of repeatedly rearranging and splitting a giant tour may be time consuming. Therefore, the FastCARP partitions the giant tour into $\lceil \sqrt{k} \rceil$ partial giant tours (PGTs), each of which is eventually split into approximately $\lceil \sqrt{k} \rceil$ vehicle routes, where k is the estimated number of routes needed in the solution. In a cyclic and overlapping manner, the algorithm now merges two adjacent PGTs. Then on the resulting merged PGT, it performs a sequence of paste, switch, shorten, and split procedures ([18], [19]) with the purpose of improving that part of the routes. After completing this process, the merged PGT is separated into two individual PGTs again. Then the whole process is repeated by merging one of the just processed PGTs with the next PGT in line.

4.5. Finalizing the Solution

The final step of the algorithm is to determine which routes to execute on each day of the planning horizon and hence to determine the total number of vehicles and the total cost.

We start by an example. Assume that $\tau = 6$ and $|\mathcal{T}| = 36$ (six weeks), and consider the district assigned to weekday 1 (Monday) and collection of fraction 1 with $l^1 = 18$. The collection frequency corresponds to three weeks. Therefore, one third of the routes in $\mathcal{R}_1^1$ can be executed in week 1 (day 1), one third in week 2 (day 7), and one third in week 3 (day 13), with this plan repeated on Mondays in weeks 4, 5, and 6. Hence, the $|\mathcal{R}_1^1|$ routes are evenly spread over l^1/τ $(18/6 = 3)$ Mondays (adjusted for rounding).

To formalize this, we consider first the non-frequent fractions $f \in \mathcal{F}, l^f \geq \tau$ with a set $\mathcal{R}_d^f$ of routes created for each of τ weekdays d. With a collection frequency of l^f days, there are l^f collection days in a cyclic plan, of which l^f/τ are the same weekday as d, and the plan is repeated $\gamma^f = |\mathcal{T}|/l^f$ times over the planning horizon. We partition the routes in $\mathcal{R}_d^f$ into l^f/τ groups, containing $\lceil |\mathcal{R}_d^f|/(l^f/\tau) \rceil$ routes in the first $(|\mathcal{R}_d^f| \mod (l^f/\tau))$ groups and $\lfloor |\mathcal{R}_d^f|/(l^f/\tau) \rfloor$

routes in the remaining $l^f/\tau - (|\mathcal{R}_d^f| \mod (l^f/\tau))$ groups. These groups are then assigned to each of the l^f/τ days of that weekday, and the routes of each group are repeated in a cyclic manner γ^f times over the planning horizon. This plan for fraction f requires $K^f = \max_{d\in\{1,\dots,\tau\}}\{\lceil|\mathcal{R}_d^f|/(l^f/\tau)\rceil\}$ vehicles and the total cost throughout the planning horizon is $C^f = \gamma^f \sum_{d\in\{1,\dots,\tau\}} C(\mathcal{R}_d^f)$, where $C(\mathcal{R}_d^f)$ represents the total cost of the routes in $\mathcal{R}_d^f$.

Next, we consider the frequent fractions $f \in \mathcal{F}, l^f < \tau$, with a set $\mathcal{R}_d^f$ of routes created for each of the first l^f weekdays. For these fractions, all $|\mathcal{R}_d^f|$ routes are executed every l^f days and therefore, a total of $|\mathcal{T}|/l^f$ times over the planning horizon. The total number of vehicles needed to service fraction f is therefore $K^f = \max_{d=1,\dots l^f}\{|\mathcal{R}_d^f|\}$, and the total cost over the planning horizon is $C^f = (|\mathcal{T}|/l^f)\sum_{d=1,\dots l^f} C(\mathcal{R}_d^f)$, where $C(\mathcal{R}_d^f)$ represents the total cost of the routes in $\mathcal{R}_d^f$.

Finally, we determine the total number of vehicles needed as $K = \sum_{f\in\mathcal{F}} K^f$, and the total routing cost over the planning horizon as $C = \sum_{f\in\mathcal{F}} C^f$.

5. Algorithm without Coordination for Comparison

Recall that the purpose of this paper is to investigate the added cost of coordination in terms of two quality measures: the number of vehicles and the total routing cost, defined as the total distance driven. In Section 4, we presented our algorithm to create coordinated solutions for our problem. In that algorithm, in order to ensure coordinated collection, we partitioned the problem into weekday districts, and aimed, via the procedure presented in Section 4.3, to minimize the number of vehicles used. We then applied the FastCARP to create routes for each waste fraction in each district, and we finally distributed these routes over the days of that weekday, while still respecting the coordination.

In order to make a comparison, we need an algorithm that solves the problem without enforcing coordination of the collections. In order to reach as fair a comparison as possible, we use the same underlying routing procedure.

When no coordination is required, we can solve the instances as a number of individual CARPs, one for each waste fraction, without creating districts, and aggregate the costs. To this end, for each waste fraction $f \in \mathcal{F}$, we use the FastCARP once with the full graph as input to obtain a set of routes $\mathcal{R}^f$.

Since fraction f needs collection with an interval of l^f days, the $|\mathcal{R}^f|$ routes are evenly spread over l^f days. As a result, we need $K = \sum_{f\in\mathcal{F}}\lceil\frac{|\mathcal{R}^f|}{l^f}\rceil$ vehicles to collect all waste fractions.

Each route for collection of f is executed $\frac{|\mathcal{T}|}{l^f}$ times during the time horizon, resulting in a total cost of $C = \sum_{f\in\mathcal{F}}\frac{|\mathcal{T}|}{l^f}C(\mathcal{R}^f)$ for collecting all fractions, where $C(\mathcal{R}^f)$ is used to denote the total cost of the routes in $\mathcal{R}^f$. This is summarized in Algorithm 4.

Algorithm 4 Route construction without coordination

for each $f \in \mathcal{F}$ **do**
  $\mathcal{R}^f \leftarrow$ FASTCARP(f)
end for
$K = \sum_{f \in \mathcal{F}} \lceil \frac{|\mathcal{R}^f|}{l^f} \rceil$
$C = \sum_{f \in \mathcal{F}} \frac{|\mathcal{T}|}{l^f} C(\mathcal{R}^f)$

6. Computational Experiments

The algorithms were implemented in C++ in MS Visual Studio Professional 2015 and executed on an Intel Xeon CPU with 12 cores running at 3.5 GHz and 64 GBs RAM. It was executed sequentially, i.e., without taking advantage of the multiple cores.

6.1. Test Instances

In the following, we describe the instances used in our experiments. For each waste fraction and each district, we allow for one minute computing time per 1,000 edges in that district requiring service of that waste fraction in the routing part of the algorithm. This means that the longest computing time including districting is only about 35 minutes, which is observed for a graph O1_E with 10,352 nodes and four waste fractions to be coordinated, two of which are frequent, using a total of 43 vehicles over a planning horizon of 12 days.

We have used the part of the benchmark data presented in [1] with homogeneous fleets for each waste fraction, with few modifications. We have made the following adjustments to the original data. 1) For all instances in sets C and E, we have made all days service days such that $\tau = 6$ instead of five. 2) To ensure that $W^f \geq q_{ij}^f$ for all f and for all (i, j), we have created new vehicles for the following four instances: F11_D, F12_D, F13_D, and S1_D. All data are available at http://www.optimization.dk/CARP/.

The data set consists of 125 instances, most of which are of very large scale, ranging up to 11,656 nodes and 12,691 edges. Underlying the 125 instances are 25 graphs: five from each of the five areas of Denmark considered in our study (F, K, N, O, and S). The total amount of waste on the edges in each of these graphs has been partitioned in different ways, and the collection intervals have been varied to create five instances based on each graph. These constitute five datasets (A,. . .,E), each containing 25 instances, one for each graph.

Table 4 shows some characteristics of the data sets after our adjustments. The instances in sets A, B, and D contain only non-frequent fractions, whereas sets

C and E also contain frequent fractions. Set C mainly differs from set B by imposing shorter frequencies for collection resulting in lower demands, and a change in τ, and the same holds for sets D and E.

	A	B	C	D	E
Number of instances	25	25	25	25	25
Time horizon (weeks)	6	6	6	12	2
Time horizon ($\mathcal{T}$) (service days)	30	30	36	60	12
Service days per week (τ)	5	5	6	5	6
Number of waste fractions	2	3	3	4	4
Intervals (l^f) (days)	10, 15	5, 10, 15	3, 12, 18	5, 10, 15, 20	2, 3, 6, 12
Intervals (weeks)	2, 3	1, 2, 3	1/2, 2, 3	1, 2, 3, 4	1/3, 1/2, 1, 2
Av. percentage of edges not req. service	36.2	36.2	36.2	36.2	36.2
Av. percentage of edges req. 1 fraction	14.0	1.2	1.2	1.2	1.2
Av. percentage of edges req. 2 fractions	49.8	13.7	13.7	0.6	0.6
Av. percentage of edges req. 3 fractions		48.9	48.9	17.9	17.9
Av. percentage of edges req. 4 fractions				44.1	44.1

Table 4: Characteristics of the five sets of C-CARP instances used in our experiments.

6.2. Results

We now present our computational results. The first part of each of Tables 7–11 provides detailed information about the instances. The first two columns provide names of the graphs and of the vehicle files. Jointly these constitute the instance. The next three columns give the number of nodes, edges, and waste fractions in the instance. Column 6 gives the number of service days in a week, while column 7 provides the number of weeks in the time horizon.

The results of our algorithm with coordination are provided in the second part of Tables 7–11. Here we give the total number of vehicles K used in the solution across all the waste fractions, the total routing cost C over all waste fractions during the whole time horizon, and the total computing time for the algorithm in seconds.

Since this paper is the first to solve this problem, we do not have a direct comparison basis. However, we analyze how the two quality measures (number of vehicles and total routing cost) are affected by the requirement that different waste fractions must be collected on the same weekday. We therefore provide the total number of vehicles K used in the solution across all the waste fractions, the total routing cost C over all waste fractions during the whole time horizon obtained with the algorithm without coordination presented in Section 5, as well as the computing time of that algorithm. These values are provided in the third part of Tables 7–11. The computing times with and without coordination are essentially the same since they depend on the number of required edges for each waste fraction.

The last part of Tables 7–11 provides the percent increase in number of vehicles and cost caused by the requirement to coordinate collections. The increase in the number of vehicles is computed as $\Delta K = 100\frac{K_{with}-K_{without}}{K_{without}}$, and the increase in cost is computed similarly. Figure 8 plots ΔK as a function of the number of nodes in the graphs, while Table 5 gives the frequency of the need for extra vehicles over all 125 instances. In total, 96 more vehicles are needed when considering all instances and all waste fractions, and extra vehicles are needed in 59 of the 125 instances. Figure 9 shows the percent increase in routing cost as a function of the number of nodes.

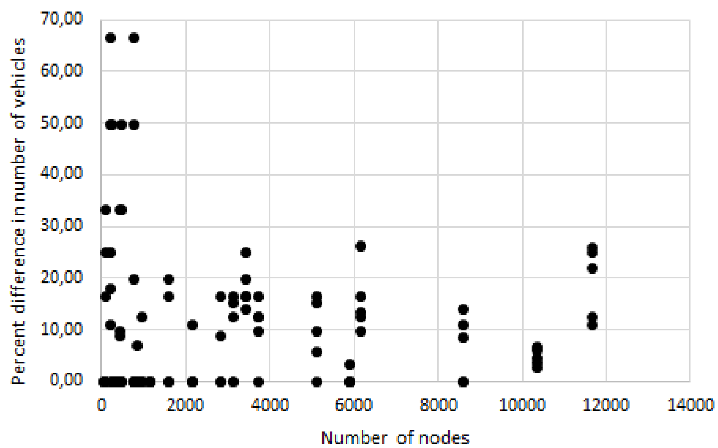


Figure 8: Percent increase in the number of vehicles as a consequence of coordination.

$K_{with} - K_{without}$	0	1	2	3	4	5	6
Frequency	66	35	17	4	1	1	1

Table 5: Frequency of the observed differences in number of vehicles.

When comparing the results with and without coordination, we observe that the routing cost with coordination increases on average by 12.4% over all 125 instances, whereas the number of vehicles increases in only 59 instances, by an average of 9.1% over all 125 instances. Figure 9 shows that the increase in routing cost caused by coordination is significantly larger for small instances than for the larger ones. We observe a cost difference of more than 10% in very few of the instances with more than 4,000 nodes. The explanation is probably that once the graph reaches a certain size, and the number of routes is likewise large, routing can still be done quite efficiently even if a coordination constraint is imposed. We observe a similar, but less clear, tendency regarding the number of vehicles in Figure 8.

Table 6 shows details of the results aggregated for each set in the left part and for each area in the right part. The four columns in each part of the table show 1) the number of the 25 instances in each set (or each area) ($| > 0|$) where coordination caused a need for extra vehicles, 2) the total number of extra vehicles needed in the sets (or areas) ($\sum$) , 3) the average percent increase in

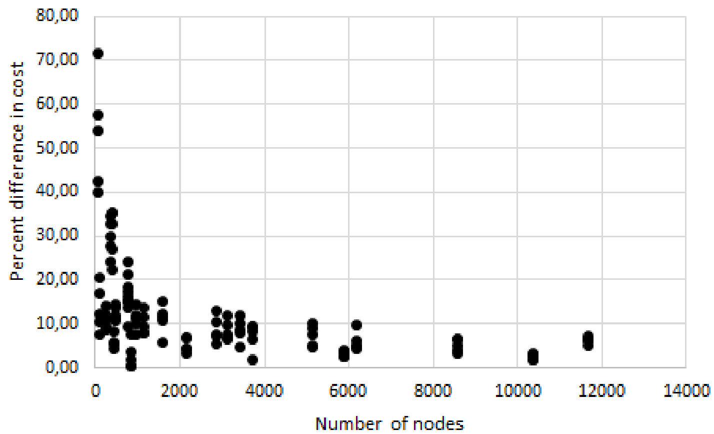


Figure 9: Percent increase in routing cost as a consequence of coordination.

	Partitioned by set					Partitioned by area			
	$\|>0\|$	$\sum$	ΔK	ΔC		$\|>0\|$	$\sum$	ΔK	ΔC
A	10	12	11.7	11.5	F	12	18	12.2	17.8
B	12	16	8.9	12.8	K	12	26	7.6	12.7
C	12	16	8.9	12.2	N	11	14	7.7	8.3
D	11	21	5.9	12.4	O	12	17	9.6	8.1
E	14	31	10.1	13.2	S	12	21	8.3	15.1
Total	59	96			Total	59	96		
Avg.			9.1	12.4	Avg.			9.1	12.4

Table 6: Average results for each dataset in the left part, and for each area in the right part. Legend: $|>0|$: Number of instances in each set (or each area) where coordination caused a need for extra vehicles; $\sum$: Total number of extra vehicles needed in the sets (or areas); ΔK: Average percent increase in the number of vehicles; ΔC: Average percent increase in routing cost.

the number of vehicles (ΔK), and 4) the average percent increase in routing cost (ΔC).

When we look across the five sets in the left of Table 6, we observe only small variations regarding the increase in routing cost. The largest changes are generally observed in set E which has both many fractions and frequent collections, both of which are factors that can complicate the solution of the problem. At the other end of the scale, we observe that set A is generally affected the least regarding routing cost. This was to be expected since the A-instances have only two fractions to coordinate. The results regarding the percent increase in vehicles also vary little across the sets, and the total number of extra vehicles needed increases, as expected, with the number of waste fractions. The set D stands out with a smaller percent increase in vehicles than the others. This may be explained by the fact that this set generally uses more vehicles per fraction than the others, as can be seen from Tables 7–11.

Comparing the five geographical areas in Table 6, we first observe that area F exhibits larger changes than the other areas. This is consistent with Figures 8 and 9 since the instances in the F area are significantly smaller than the other instances, the largest having less than 1,000 nodes. Among the other four areas, areas K and S show a larger increase in routing cost. A possible explaining factor for this behaviour may lie in the non-convex shape of these two areas, which exacerbates the consequences of poor routing decisions.

7. Conclusions

We have considered a multi-period garbage collection problem involving several garbage types called *fractions*, such as organic waste, paper and cardboard, glass and metal, and plastic. This study was motivated by a real-life problem arising in Denmark. We have obtained data for six counties, two of which are rural, two are semi-rural (and were considered as a single area in our experiments), and two are urban. The instances sizes are very large and can reach 11,656 nodes and 12,691 edges. Because of the nature of the fractions and variations in volumes, not all of them have the same frequency. The purpose of the paper was to assess the added cost in terms of traveled distance and vehicle fleet size of coordinating these collections such that each household would always have its collection on the same day of the week.

Since the problem is of very large scale, we have developed an efficient constructive heuristic that does not resort to the application of computationally expensive exchange mechanisms. Our heuristic was made up of four phases: 1) collection districts, each corresponding to a day of the week, are first created; 2) the districts are then assigned to specific weekdays based on a closeness criterion; 3) they are then balanced in order to make a more efficient use of the vehicles; 4) collection routes are then created for each district and each waste fraction by means of the FastCARP heuristic. The objective minimized in this problem is hierarchical, the fleet size being more important than the routing cost.

The heuristic was extensively tested over 125 instances made up of 25 graphs for each of the five counties considered in the study. We show that coordinating the collection days results in a routing cost increase of 12.4% and in an increase of 9.1% in the number of vehicles. The number of vehicles increased in only 59 of all instances. We observed a smaller cost increase in the larger instances, and a larger increase in the instance set that has both many fractions and frequent collections, both of which are complicating factors. The instance sets that use more vehicles per fraction are those in which the percent increase in the number of vehicles is the smallest. Comparing the five geographical areas, we found that the cost increase is larger in the smaller areas and in those that have irregular shapes. Deciding whether such cost increases are acceptable in order to provide better service for the citizens is left to the counties.

Acknowledgements

This project was funded by the Danish Council for Independent Research - Social Sciences. Project 'Transportation issues related to waste management' [grant number 4182-00021] and by the Natural Sciences and Engineering Research Council of Canada [grant number 2015-06189]. This support is gratefully acknowledged. Thanks are due to the Editor, the Associate Editor, and the referees for their support and valuable comments.

References

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

[2] E. Bartolini, J.-F. Cordeau, G. Laporte, Improved lower bounds and exact algorithm for the capacitated arc routing problem, Mathematical Programming 137 (1–2) (2013) 409–452.

[3] C. Prins, The capacitated arc routing problem: Heuristics, in: A. Corberán, G. Laporte (Eds.), Arc Routing: Problems, Methods, and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2015.

[4] S. Wøhlk, G. Laporte, A fast heuristic for very large-scale capacitated arc routing problems, Journal of the Operational Research Society, 69 (12) (2018) 1877–1887.

[5] L. Muyldermans, D. Cattrysse, D. Van Oudheusden, District design for arc-routing applications, Journal of the Operational Research Society 54 (11) (2003) 1209–1221.

[6] M.C. Mourão, A. Nunes, C. Prins, Heuristic methods for the sectoring arc routing problem, European Journal of Operational Research 196 (3) (2009) 856–868.

[7] A. Butsch, J. Kalcsics, G. Laporte, Districting for arc routing, INFORMS Journal on Computing 26 (4) (2014) 809–824.

[8] L. Muyldermans, G. Pang, On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm, European Journal of Operational Research 206 (1) (2010) 93–103.

[9] H. Zbib, Variants of the path scanning construction heuristic for the no-split multi-compartment capacitated arc routing problem, Working Paper, Aarhus University (2017).

[10] H. Zbib, S. Wøhlk, A multi-move chain descent algorithm for large-scale arc routing problems in curbside waste collection, submitted (2018).

[11] M.C. Mourão, L. Amado, Heuristic method for a mixed capacitated arc routing problem: A refuse collection application, European Journal of Operational Research 160 (1) (2005) 139–153.

[12] G. Ghiani, M.C. Mourão, L. Pinto, D. Vigo, Routing in waste collection applications, in: A. Corberán, G. Laporte (Eds.), Arc Routing: Problems, Methods, and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2015.

[13] C. Mora, R. Manzini, M. Gamberi, A. Cascini, Environmental and economic assessment for the optimal configuration of a sustainable solid waste collection system: a kerbside case study, Production Planning & Control 25 (9) (2014) 737–761.

[14] A.M. Rodrigues, J.S. Ferreira, Waste collection routing – limited multiple landfills and heterogeneous fleet, Networks 65 (2) (2015) 155–165.

[15] X. Bing, J.M. Bloemhof, T.R.P. Ramos, A.P. Barbosa-Povoa, C.Y. Wong, J.G.A.J. van der Vorst, Research challenges in municipal solid waste logistics management, Waste Management 48 (2016) 584–592.

[16] A. Corberán, G. Laporte (Eds.), Arc Routing: Problems, Methods, and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2015.

[17] M.C. Mourão, L.S. Pinto, An updated annotated bibliography on arc routing problems, Networks 70 (3) (2017) 144–194.

[18] A. Hertz, G. Laporte, P. Nanchen-Hugo, Improvement procedures for the undirected rural postman problem, INFORMS Journal on Computing 11 (1) (1999) 53–62.

[19] A. Hertz, G. Laporte, M. Mittaz, A tabu search heuristic for the capacitated arc routing problem, Operations Research 48 (1) (2000) 129–135.

Characteristics of the data							With coordination			Without coordination			Change %	
Graph	Veh.	$\|\mathcal{N}\|$	$\|\mathcal{E}\|$	$\|\mathcal{F}\|$	τ	$\|\mathcal{T}\|/\tau$	K	C	Time(s)	K	C	Time(s)	ΔK	ΔC
F1_A	A5	812	1124	2	5	6	12	2401325	90.7	12	2388166	90.6	0.0	0.6
F10_A	A4	415	565	2	5	6	7	617137	43.5	7	581697	43.5	0.0	6.1
F11_A	A3	191	267	2	5	6	5	197028	20.0	3	175792	20.0	66.7	12.1
F12_A	A2	80	110	2	5	6	4	106199	8.3	4	96107	8.2	0.0	10.5
F13_A	A2	26	33	2	5	6	2	23987	1.8	2	17114	2.2	0.0	40.2
K1_A	A5	11656	12691	2	5	6	10	18613350	880.0	8	17328677	890.0	25.0	7.4
K10_A	A4	5102	5518	2	5	6	7	6994405	371.8	6	6347496	371.7	16.7	10.2
K11_A	A4	3114	3361	2	5	6	4	4564074	220.4	4	4150641	217.1	0.0	10.0
K12_A	A2	1132	1221	2	5	6	3	2071799	75.1	3	1888493	75.2	0.0	9.7
K13_A	A1	394	422	2	5	6	2	890754	27.8	2	657314	27.8	0.0	35.5
N1_A	A4	8573	9761	2	5	6	10	23222524	689.0	10	22311251	696.0	0.0	4.1
N10_A	A4	3698	4187	2	5	6	5	6250800	291.3	5	5765363	292.3	0.0	8.4
N11_A	A3	2142	2419	2	5	6	3	3405377	182.7	3	3295233	183.0	0.0	3.3
N12_A	A1	930	1040	2	5	6	6	2106575	81.4	6	1955748	81.3	0.0	7.7
N13_A	A2	454	502	2	5	6	3	716054	42.4	2	638472	42.3	50.0	12.2
O1_A	A5	10352	11943	2	5	6	17	14744723	1040.4	16	14281659	1038.6	6.3	3.2
O10_A	A5	5882	6982	2	5	6	12	8334158	564.7	12	8041716	567.7	0.0	3.6
O11_A	A4	2822	3281	2	5	6	7	4273669	248.8	6	3976822	248.0	16.7	7.5
O12_A	A2	761	852	2	5	6	4	1070036	62.2	4	975441	62.1	0.0	9.7
O13_A	A1	228	247	2	5	6	3	310397	19.9	2	276880	19.8	50.0	12.1
S1_A	A3	6149	7110	2	5	6	7	7889720	349.4	6	7547090	352.7	16.7	4.5
S10_A	A3	3404	3921	2	5	6	5	4874394	200.0	4	4485731	200.7	25.0	8.7
S11_A	A1	1564	1805	2	5	6	6	2580274	87.9	5	2433250	87.8	20.0	6.0
S12_A	A1	755	866	2	5	6	3	1197031	39.5	3	962642	39.5	0.0	24.3
S13_A	A2	322	374	2	5	6	2	479302	18.6	2	368330	18.6	0.0	30.1

Table 7: Detailed results for data set A.

Characteristics of the data							With coordination			Without coordination			Change %	
Graph	Veh.	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	τ	$\lvert\mathcal{T}\rvert/\tau$	K	C	Time(s)	K	C	Time(s)	ΔK	ΔC
F1_B	B5	812	1124	3	5	6	18	3954765	131.5	18	3931002	131.4	0.0	0.6
F10_B	B4	415	565	3	5	6	11	1093377	62.2	10	1047282	62.2	10.0	4.4
F11_B	B3	191	267	3	5	6	6	342678	28.7	4	310819	28.6	50.0	10.3
F12_B	B2	80	110	3	5	6	5	155263	12.0	4	144128	11.9	25.0	7.7
F13_B	B2	26	33	3	5	6	3	46917	3.2	3	30401	3.2	0.0	54.3
K1_B	B5	11656	12691	3	5	6	22	44324717	1399.7	18	41496404	1449.3	22.2	6.8
K10_B	B4	5102	5518	3	5	6	13	16458253	598.1	13	15085359	600.7	0.0	9.1
K11_B	B4	3114	3361	3	5	6	9	10947338	358.1	8	9771948	356.5	12.5	12.0
K12_B	B2	1132	1221	3	5	6	3	4513520	123.4	3	4035806	123.5	0.0	11.8
K13_B	B1	394	422	3	5	6	3	1953171	44.8	3	1441544	44.7	0.0	35.5
N1_B	B4	8573	9761	3	5	6	17	46227612	1079.0	17	43889725	1092.9	0.0	5.3
N10_B	B4	3698	4187	3	5	6	9	13567568	459.8	8	12369617	461.5	12.5	9.7
N11_B	B3	2142	2419	3	5	6	5	7035739	278.9	5	6573564	279.9	0.0	7.0
N12_B	B1	930	1040	3	5	6	7	3846423	123.3	7	3361446	123.5	0.0	14.4
N13_B	B2	454	502	3	5	6	4	1255040	64.2	3	1100487	64.2	33.3	14.0
O1_B	B5	10352	11943	3	5	6	36	32315804	1572.0	35	31630511	1572.6	2.9	2.2
O10_B	B5	5882	6982	3	5	6	25	18696113	853.4	25	17923742	859.7	0.0	4.3
O11_B	B4	2822	3281	3	5	6	12	8921980	374.6	11	8445946	375.0	9.1	5.6
O12_B	B2	761	852	3	5	6	6	2052871	94.3	5	1779378	94.4	20.0	15.4
O13_B	B1	228	247	3	5	6	3	574172	30.1	3	523543	30.0	0.0	9.7
S1_B	B3	6149	7110	3	5	6	11	16942833	578.5	10	16141139	592.3	10.0	5.0
S10_B	B3	3404	3921	3	5	6	8	10710252	335.2	7	9566159	335.9	14.3	12.0
S11_B	B1	1564	1805	3	5	6	7	5235439	145.8	7	4713532	145.7	0.0	11.1
S12_B	B1	755	866	3	5	6	3	2308505	64.0	3	1948421	64.0	0.0	18.5
S13_B	B2	322	374	3	5	6	3	1041566	29.2	3	784412	29.1	0.0	32.8

Table 8: Detailed results for data set B.

Characteristics of the data							With coordination			Without coordination			Change %	
Graph	Veh.	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	τ	$\lvert\mathcal{T}\rvert/\tau$	K	C	Time(s)	K	C	Time(s)	ΔK	ΔC
F1_C	C5b	812	1124	3	6	6	15	4707609	131.5	14	4617718	131.5	7.1	1.9
F10_C	C4b	415	565	3	6	6	8	1485360	62.2	8	1371270	62.2	0.0	8.3
F11_C	C3b	191	267	3	6	6	5	501080	28.7	4	446089	28.7	25.0	12.3
F12_C	C2b	80	110	3	6	6	4	231293	11.9	4	197708	11.9	0.0	17.0
F13_C	C2b	26	33	3	6	6	3	68210	3.2	3	43229	3.2	0.0	57.8
K1_C	C5b	11656	12691	3	6	6	18	61238726	1375.9	16	57460406	1446.1	12.5	6.6
K10_C	C4b	5102	5518	3	6	6	11	22064792	604.2	10	21044740	601.1	10.0	4.8
K11_C	C4b	3114	3361	3	6	6	7	14753952	354.2	6	13803396	356.9	16.7	6.9
K12_C	C2b	1132	1221	3	6	6	3	6588276	123.2	3	6095096	124.0	0.0	8.1
K13_C	C1b	394	422	3	6	6	3	2733150	44.7	3	2146736	44.7	0.0	27.3
N1_C	C4b	8573	9761	3	6	6	16	59384553	1070.8	14	55755049	1098.2	14.3	6.5
N10_C	C4b	3698	4187	3	6	6	7	18109415	459.3	6	17780633	460.3	16.7	1.8
N11_C	C3b	2142	2419	3	6	6	5	10108535	278.4	5	9442578	279.3	0.0	7.1
N12_C	C1b	930	1040	3	6	6	7	5107791	123.3	7	4563054	123.5	0.0	11.9
N13_C	C2b	454	502	3	6	6	3	1755123	64.2	3	1573689	64.2	0.0	11.5
O1_C	C5b	10352	11943	3	6	6	31	44686951	1570.1	29	43418983	1575.9	6.9	2.9
O10_C	C5b	5882	6982	3	6	6	21	24533144	860.2	21	23922154	859.9	0.0	2.6
O11_C	C4b	2822	3281	3	6	6	10	12610972	374.7	10	11383977	376.2	0.0	10.8
O12_C	C2b	761	852	3	6	6	5	2934546	94.2	3	2522976	94.5	66.7	16.3
O13_C	C1b	228	247	3	6	6	3	791417	30.0	3	727447	30.0	0.0	8.8
S1_C	C3b	6149	7110	3	6	6	9	24751924	577.9	8	23289341	589.6	12.5	6.3
S10_C	C3b	3404	3921	3	6	6	7	14505336	334.9	6	13410299	336.4	16.7	8.2
S11_C	C1b	1564	1805	3	6	6	7	7577122	145.7	6	6577408	145.8	16.7	15.2
S12_C	C1b	755	866	3	6	6	3	3270445	63.9	3	2833403	64.0	0.0	15.4
S13_C	C2b	322	374	3	6	6	3	1512569	29.1	3	1181300	29.1	0.0	28.0

Table 9: Detailed results for data set C.

Characteristics of the data							With coordination			Without coordination			Change %	
Graph	Veh.	$\|\mathcal{N}\|$	$\|\mathcal{E}\|$	$\|\mathcal{F}\|$	τ	$\|\mathcal{T}\|/\tau$	K	C	Time(s)	K	C	Time(s)	ΔK	ΔC
F1_D	D4	812	1124	4	5	12	20	8249938	148.7	20	7935147	148.6	0.0	4.0
F10_D	D3	415	565	4	5	12	12	2227101	73.0	11	2112499	73.0	9.1	5.4
F11_D	D2-C	191	267	4	5	12	13	1037292	34.2	11	922890	34.2	18.2	12.4
F12_D	D1-B	80	110	4	5	12	8	394254	14.2	6	350423	14.4	33.3	12.5
F13_D	D1-B	26	33	4	5	12	4	109389	3.2	4	76752	3.9	0.0	42.5
K1_D	D4	11656	12691	4	5	12	30	112076288	1912.5	27	105507020	2004.6	11.1	6.2
K10_D	D3	5102	5518	4	5	12	18	40552422	823.4	17	37612885	832.2	5.9	7.8
K11_D	D2	3114	3361	4	5	12	16	31952347	494.1	16	29605868	494.5	0.0	7.9
K12_D	D1	1132	1221	4	5	12	5	11328959	171.4	5	9935390	172.0	0.0	14.0
K13_D	D1	394	422	4	5	12	4	4443555	61.6	4	3339915	61.6	0.0	33.0
N1_D	D3	8573	9761	4	5	12	25	114391864	1459.2	23	110656848	1490.2	8.7	3.4
N10_D	D3	3698	4187	4	5	12	11	33014866	627.4	10	30105246	630.1	10.0	9.7
N11_D	D2	2142	2419	4	5	12	11	19347509	374.8	11	18495565	375.7	0.0	4.6
N12_D	D1	930	1040	4	5	12	9	8473102	165.1	8	7604034	165.4	12.5	11.4
N13_D	D1	454	502	4	5	12	5	3289028	86.0	5	2873883	86.2	0.0	14.4
O1_D	D4	10352	11943	4	5	12	54	81748354	2081.0	52	79120550	2102.9	3.8	3.3
O10_D	D4	5882	6982	4	5	12	36	46908913	1143.8	36	45600014	1152.8	0.0	2.9
O11_D	D3	2822	3281	4	5	12	16	22480164	502.2	16	20894297	503.6	0.0	7.6
O12_D	D2	761	852	4	5	12	7	4867828	126.4	7	4276153	126.5	0.0	13.8
O13_D	D1	228	247	4	5	12	4	1362899	40.3	4	1193420	40.2	0.0	14.2
S1_D	D2-B	6149	7110	4	5	12	25	53145421	808.0	22	50567563	829.7	13.6	5.1
S10_D	D2	3404	3921	4	5	12	18	32561313	467.2	15	31061867	470.9	20.0	4.8
S11_D	D1	1564	1805	4	5	12	8	12166313	203.4	8	10875143	204.3	0.0	11.9
S12_D	D1	755	866	4	5	12	4	5541355	88.4	4	4559321	88.5	0.0	21.5
S13_D	D1	322	374	4	5	12	4	2621767	39.7	4	1947317	39.7	0.0	34.6

Table 10: Detailed results for data set D.

Characteristics of the data							With coordination			Without coordination			Change %	
Graph	Veh.	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	τ	$\lvert\mathcal{T}\rvert/\tau$	K	C	Time(s)	K	C	Time(s)	ΔK	ΔC
F1_E	E4b	812	1124	4	6	2	16	2238077	148.7	16	2077676	148.7	0.0	7.7
F10_E	E3b	415	565	4	6	2	12	816671	73.0	9	734129	73.0	33.3	11.2
F11_E	E2b	191	267	4	6	2	10	335000	34.2	9	299806	34.2	11.1	11.7
F12_E	E1b	80	110	4	6	2	7	151859	14.4	6	125631	14.4	16.7	20.9
F13_E	E1b	26	33	4	6	2	4	55024	3.9	4	32005	3.9	0.0	71.9
K1_E	E4b	11656	12691	4	6	2	29	42966426	1911.4	23	40801377	2038.9	26.1	5.3
K10_E	E3b	5102	5518	4	6	2	15	15766300	833.9	13	14974819	836.1	15.4	5.3
K11_E	E2b	3114	3361	4	6	2	15	11417656	494.1	13	10719154	497.7	15.4	6.5
K12_E	E1b	1132	1221	4	6	2	5	5043402	171.5	5	4663654	172.4	0.0	8.1
K13_E	E1b	394	422	4	6	2	4	1969899	61.7	4	1608421	61.7	0.0	22.5
N1_E	E3b	8573	9761	4	6	2	20	37461703	1466.6	18	35060302	1494.6	11.1	6.8
N10_E	E3b	3698	4187	4	6	2	9	13386191	630.0	8	12555732	638.6	12.5	6.6
N11_E	E2b	2142	2419	4	6	2	10	7832986	375.7	9	7486471	377.7	11.1	4.6
N12_E	E1b	930	1040	4	6	2	7	3316374	165.2	7	3007867	165.9	0.0	10.3
N13_E	E1b	454	502	4	6	2	5	1273235	86.0	5	1145434	86.1	0.0	11.2
O1_E	E4b	10352	11943	4	6	2	45	30928479	2103.6	43	30367398	2122.2	4.7	1.8
O10_E	E4b	5882	6982	4	6	2	31	16876108	1154.0	30	16256512	1154.9	3.3	3.8
O11_E	E3b	2822	3281	4	6	2	14	8587787	502.6	14	7597222	506.4	0.0	13.0
O12_E	E2b	761	852	4	6	2	6	2071192	126.4	4	1797778	126.8	50.0	15.2
O13_E	E1b	228	247	4	6	2	4	577311	40.3	4	519967	40.4	0.0	11.0
S1_E	E2b	6149	7110	4	6	2	24	19955327	811.5	19	18180766	831.0	26.3	9.8
S10_E	E2b	3404	3921	4	6	2	14	11863491	468.9	12	10767274	473.9	16.7	10.2
S11_E	E1b	1564	1805	4	6	2	7	5005755	203.6	7	4456864	204.4	0.0	12.3
S12_E	E1b	755	866	4	6	2	4	2399297	88.4	4	2039757	88.5	0.0	17.6
S13_E	E1b	322	374	4	6	2	4	1124471	39.7	4	905298	39.7	0.0	24.2

Table 11: Detailed results for data set E.