EFFICIENT WASTE COLLECTION BY MEANS OF ASSIGNMENT PROBLEMS

Kostanca Katragjini, Federico Perea and Rubén Ruiz

Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática
Universitat Politècnica de València, Camino de Vera s/n, 46021 Valencia, Spain

Keywords: Waste collection, Linear programming, Municipal solid waste management, Assignment.

Abstract: This paper studies a municipal waste collection problem, whose ultimate goal is to find an assignment of containers to service days and posterior routing so that the total distance covered by the waste collection trucks is minimized. Due to the complexity of this problem, we divide it into several subproblems. The focus of this paper is on the first problem, in which it must be decided which containers must be collected on each service day so that the containers collected on the same day are as close to each other as possible. In a second phase (out of the scope of this paper) the routes followed by the collection vehicles at each service day are optimized. Such separation of the problem is needed as a result of the sheer size and complexity of the real setting. Despite this separation into smaller subproblems, efficient procedures must be designed in order to solve real-sized instances. A computational experience over a number of instances shows the applicability of our methods.

1 INTRODUCTION

This paper shows the initial results obtained when working on a problem proposed by a municipality regarding their waste management. More specifically, the municipality wants a support system able to decide when and how to collect their waste containers so that the associated costs are minimized. Other requirements are that such support system should be based on free software and that a solution should be given in less than five minutes.

Waste collection (WC) is an essential problem in many cities. Due to the increasing concerns about pollution, and the ever raising fuel prices, an efficient planning is necessary for both saving money and protecting the environment. WC can be seen as part of the municipal solid waste (MSW) management problem, in which several processes are considered: waste collection, transportation, treatment and disposal, (Wilson, 1985). In the last decade, there has been a trend to study the MSW management problem taking into account the uncertainty that some of the involved parameters have, see for instance (Huang et al., 2002; Li et al., 2008) and the references therein.

The WC problem can be defined over a road network of a given city. On such network, the different elements of the problem are coded:

- one or more depots, where vehicles are stationed.
- landfills and disposal plants, where waste has to be brought.
- waste containers, which are usually specialized for different types of waste.
- a fleet of waste collection vehicles, potentially heterogeneous.
- service frequency.
- and many other side constraints.

The actual problem faced asks for designing the routes of the available vehicles, in order to collect all specified containers, satisfying capacity and container compatibility constraints with the objective of minimizing the total travel distance and costs. In (Golden et al., 2002), the waste collection problem is considered as an application of the vehicle routing problem. (Maniezzo, 2004) goes a bit further and formulates the WC problem as a capacitated vehicle routing problem. In (Angelilli and Speranza, 2002) it is discussed the importance of good estimates for the operating costs in the corresponding vehicle routing problem, which is applied to model two case studies. The authors also consider work shifts and other important constraints. More complicated models arise when other constraints are taken into account, such as

Katragjini K., Perea F. and Ruiz R..
EFFICIENT WASTE COLLECTION BY MEANS OF ASSIGNMENT PROBLEMS.
DOI: 10.5220/0003745102830292
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 283-292
ISBN: 978-989-8425-97-3

drivers' lunch breaks, see e.g. (Kim et al., 2006; Benjamin and Beasley, 2010) and many others. For example, different types of waste might be present, sometimes collected with different trucks which employ different technologies. Examples of different types of waste are glass, plastic, containers, cardboard, dangerous waste (such as fluorescent light bulbs, used oil, batteries, etc.), organic waste, etc. Moreover, WC complicates further if we consider the fact that the waste truck, once full, has to do a potentially long trip to the treatment plant or landfill and then it has to go back to a potentially different point in the route to continue with the collection. Additionally, all constraints typical to labor scheduling problems also appear, as the maximum driving time and working time of the waste collection operators must be observed. As a result of all of the above, the WC can be considered as a very complicated and difficult vehicle routing problem, or better as a rich periodic vehicle routing problem (PVRP). For early references on PVRP and WC related problems the reader is referred to (Beltrami and Bodin, 1974), who use the PVRP to model a municipal waste collection problem, and (Russell and Igo, 1979), who assign customer demand points to days of the week so that a certain node routing problem is solved more efficiently. For a more recent reference see for instance (Coene et al., 2010) and the references therein.

Additionally, the company wanted us to study a specific generalization of the WC problem. In this setting, there are two types of waste that are collected simultaneously, typically cardboard and general waste, as these two types of waste are the most frequent and abundant. In order to do so, a special truck with two compartments is employed. When the truck reaches a location or node, it collects the cardboard container (if present) and the general waste (usually always present). From now on, following the usual nomenclature in WC problems, the two types of waste are referred to as *fractions*.

Due to the high complexity of the WC problem, a large variety of heuristics are present in the literature. In (Teixeira et al., 2004), a constructive heuristic divided into three phases, namely definition of geographic zones, definition of the waste type to collect each day, and definition of routes, is presented to solve a urban recyclable waste collection problem. A clustering-based waste collection algorithm is presented in (Kim et al., 2006). (Baustista et al., 2008) model a WC problem as an capacitated arc routing problem with some specific characteristics, which is transformed into a node routing problem and solved by means of an ant colony heuristic. Three metaheuristics are presented in (Benjamin and Beasley, 2010). The authors state that their algorithms perform better in terms of quality of solutions than other algorithms previously presented in the literature.

The company wanted this problem to be solved (with around 1000 different collection sites) in less than five minutes by means of free software. Time constraints are due to the daily operations which needed fast response times and free software was needed due to budget limitations. Our experience showed that it was impossible to do that (we could solve to optimality instances consisting of 10 locations in around a minute by means of CPLEX, but such computational time explodes as the number of locations increases). Therefore, we had to propose the following partition of the WC setting into two different problems:

1. Assignment: in a first step we would find an assignment of locations to service days so that all locations collected on the same day are close to each other and the amount of waste collected per day is relatively constant.
2. Routing: the routing problem for each service day would be solved on a second step. This problem is not studied in this paper as it represents a very complex vehicle routing problem. Additionally, the problem may involve hundreds of nodes (up to a thousand) and as much as 50 waste collection trucks.

This paper shows the progress made on the first step. In Section 2, this assignment problem is modeled as a mixed integer linear programming problem. Section 3 introduces procedures aiming at solving our assignment part of the WC problem efficiently. Such procedures are tested and compared in Section 4. Some additional techniques, yet to be tested, are presented in Section 5. The paper closes with some conclusions and pointers to future work.

2 ASSIGNING LOCATIONS TO SERVICE DAYS

In this section we propose a mixed integer linear programming (MILP) model that decides which locations should be collected on each service day, so that no container overflows, the amount of waste collected per service day is relatively constant, and the locations visited on each service day are as close to each other as possible, over a weekly planning horizon. This last objective has been treated by minimizing the sum of the diameters of each day, as we will explain later. A service day is defined as a day in which some waste is collected. There are 7 days in a week but there does not need to be as many service days. For example, busy streets with lots of population or businesses nearby have their containers serviced every day, potentially 7 days per week. Conversely, sparsely populated areas might have containers serviced twice or even just once per week. Note that the problem also includes a given number of service days per week.

Our problem has the following input data:

- Let $i = 1,\dots,I$ be the locations where the containers are placed, $j = 1,\dots,7$ the possible service days (Monday to Sunday), and $k = 1,2$ waste types, from now on denoted by fraction 1 and fraction 2, respectively.
- Let (α_i, β_i) be the coordinates of location i on the plane, $i = 1,\dots,I$.
- There are n_i^k containers of fraction k at location i, for all $i = 1,\dots,I$, $k = 1,2$. In reality, there are many more waste types (organic waste, paper, plastic, hazardous, and others) but since the collection technologies differ, the problems can be solved separately in different runs. Therefore, only two fractions are considered.
- Each container of fraction k receives g_k kilograms of waste every day, and has a maximum capacity of c_k, $k = 1,2$.
- The company wants to collect fraction k s_k days per week, $k = 1,2$, with $s_k \in \{1,\dots,7\}$. These are the desired service days.
- Each fraction k container must be collected f_k days per week. Note that $f_k \leq s_k$. This parameter is the so called frequency of service.
- If location i is visited to collect fraction 2, then fraction 1 on location i must be collected as well. This implies that the actual number of service days is s_1.

The last point above is another specific constraint. The company wants the "minor" fraction to be collected each time the "major" fraction is collected. In other words, when the general waste fraction is collected by a truck in a given location, the cardboard waste in the same location (if present) must be collected as well.

It is also very important to note the difference between the service days of each fraction s_k and the frequency for each fraction f_k. Let us advance an example. The company wants to service an area with maximum 4 service days in a week. This means that $s_1 = s_2 = 4$. In other words, waste collection trucks should operate only four days during the week. The frequency for a different fraction is a different matter, $f_1 = 3$ means that the containers of the first fraction have to be collected three times per week. Since not all containers have to be collected in the same service days, the problem is therefore to distribute containers in service days so that every container is collected three times per week and that there are no more than four days with service.

2.1 Preprocessing: Timetables

First, given the f_k values, the set of feasible timetables is calculated for a given collection site. Each timetable determines on which days a location is visited, and they will be assigned to collection sites. From this information, and from the amount of waste that arrives at each container per day (given by parameter g_k), the amount of waste collected each day can be easily calculated. A timetable is defined as a matrix $p \in \mathbb{R}^{7\times 4}$ such that:

- $p_{j,k} = 1$ if timetable p determines that on day j, fraction k must be collected, $j = 1,\dots,7, k = 1,2$, zero otherwise.
- $p_{j,k+2}$ is the amount of fraction k collected on day j if the timetable p is followed, $j = 1,\dots,7, k = 1,2$. Note that, if $p_{j,k} = 0$, then $p_{j,k+2} = 0$ too. Note as well that $p_{j,k+2}$ only depends on g_k and on the last day before j that fraction k was collected.

Each feasible timetable p must satisfy:

1. $p_{j,2} \leq p_{j,1},\ j = 1,\dots,7$ (if fraction 2 is collected on day j, so should fraction 1).
2. $\sum_{j=1}^{7} p_{j,k} = f_k,\ k = 1,2$ (for every collection site, fraction k must be collected f_k times per week).
3. $p_{j,k+2} \leq c_k,\ j = 1,\dots,7, k = 1,2$ (containers cannot overflow their capacity).

Note that the list of feasible timetables only depends on f_k, and on the ratio c_k/g_k, for $k = 1,2$. Besides, the company may want to add some extra constraints, such that no container must be collected on two consecutive days. We will consider that there are m feasible timetables $\{p^1,\dots,p^m\}$.

Example 1. *As an example, consider a situation in which* $s_1 = s_2 = 7, f_1 = 2, f_2 = 1, g_1 = g_2 = 10, c_1 = 40, c_2 = 80$. *Consider the following two timetables:*

$$p = \begin{pmatrix} 1 & 1 & 30 & 70 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ p' = \begin{pmatrix} 1 & 1 & 20 & 70 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Timetable p determines that the fraction 1 is to be collected on Mondays and on Fridays, and fraction 2 *on*

Mondays. Timetable p' determines that fraction 1 *is to be collected on Mondays and on Saturdays, and fraction* 2 *on Mondays. It is easy to see that p is a feasible timetable, whereas p' is not, since a fraction* 1 *container will overflow on Saturdays before the collection (amount of garbage to be collected is 50, whereas the container's maximum capacity for fraction* 1 *is 40).*

Timetables are enumerated, note that in principle there are $\binom{7}{f_1}\binom{f_1}{f_2}$ possible timetables. Nevertheless, due to containers' capacity and estimated amount of waste received per day, many of these timetables are unfeasible and therefore removed. Note that this procedure could be very well extended and generalized so to work at each location, i.e., each location could have a different capacity and a different generation of garbage. However, the company did not have as much precise and detailed data for such an extension.

2.2 Variables

Given the feasible timetables, all that is left to do is to assign one timetable to each location so that the given objective is optimized. For this assignment problem, we employ the following variables:

- $x_i^\ell = 1$ if location i is assigned to timetable p^ℓ, zero otherwise.
- $y_j^k = 1$ if on day j fraction k is collected, zero otherwise.
- v is an internal variable, which is used to balance the amount of waste collected per day.
- a_j, b_j are the first and second coordinates of the center of day j, respectively.
- $r_j \geq 0$ is the radius of day j.

2.3 Constraints

The constraints of our model are:

- Exactly one timetable for each location:

$$\sum_{\ell=1}^{m} x_i^\ell = 1;\ i = 1,...,I. \tag{1}$$

- The following constraints force the number of service days in which fraction k is collected to be s_k:

$$\sum_{j=1}^{7} y_j^k = s_k;\ k = 1,2. \tag{2}$$

$$y_j^k \leq \sum_{i=1}^{I}\sum_{\ell=1}^{m} x_i^\ell p_{j,k}^\ell;\ j = 1,...,7; k = 1,2. \tag{3}$$

$$x_i^\ell p_{j,k}^\ell \leq y_j^k;\ i = 1,...,I;\ j = 1,...,7; \quad \ell = 1,...,m; k = 1,2. \tag{4}$$

Note that Equations (3) and (4) force that $y_j^k = 1$ if and only if at least one collection site is visited to collect fraction k on day j.

- Since the company wants the amount of waste collected per day to be relatively constant, we have to add:

$$-(1-y_j^1)M + (1-\varepsilon)v \leq \sum_{i=1}^{I}\sum_{\ell=1}^{m}\sum_{k=1}^{2} x_i^\ell p_{j,k+2}^\ell n_i^k, \quad j = 1,\ldots,7, \tag{5}$$

$$\sum_{i=1}^{I}\sum_{\ell=1}^{m}\sum_{k=1}^{2} x_i^\ell p_{j,k+2}^\ell n_i^k \leq (1+\varepsilon)v + (1-y_j^1)M, \quad j = 1,\ldots,7, \tag{6}$$

where $\varepsilon \in [0,1]$ is a small tolerance, and M is a large enough positive number. These two sets of constraints force that the amount of collected waste on a service day (when $y_j^1 = 1$) is in the interval $[(1-\varepsilon)v, (1+\varepsilon)v]$.

- The radius of a given day j is defined as:

$$\min_{(a_j,b_j)\in\mathbb{R}^2}\ \max_{i \text{ collected on day } j} d((a_j,b_j),i),$$

where $d((a_j,b_j),i)$ is the distance from (a_j,b_j) to location i. The point (a_j,b_j) defining the radius of a given day j is called the *center* of day j. In order to keep the linearity of the problem we will use the Manhattan distance. The following constraints force that all locations collected on day j are less than r_j far from the center (a_j,b_j) of that day, using the Manhattan distance:

$$-M(1-\sum_{\ell=1}^{m} x_i^\ell p_{j,1}^\ell) + |\alpha_i - a_j| + |\beta_i - b_j| \leq r_j, \quad i = 1,...,I, j = 1,...,7.$$

These constraints can easily be made linear by dividing them into the four possible cases.

$$-M(1-\sum_{\ell=1}^{m} x_i^\ell p_{j,1}^\ell) + (\alpha_i - a_j) + (\beta_i - b_j) \leq r_j, \tag{7}$$

$$-M(1-\sum_{\ell=1}^{m} x_i^\ell p_{j,1}^\ell) + (\alpha_i - a_j) - (\beta_i - b_j) \leq r_j, \tag{8}$$

$$-M(1-\sum_{\ell=1}^{m} x_i^\ell p_{j,1}^\ell) - (\alpha_i - a_j) + (\beta_i - b_j) \leq r_j, \tag{9}$$

$$-M(1-\sum_{\ell=1}^{m} x_i^\ell p_{j,1}^\ell) - (\alpha_i - a_j) - (\beta_i - b_j) \leq r_j, \tag{10}$$

$$i = 1,...,I, j = 1,...,7.$$

2.4 Objective

The objective is to minimize the average radius of each day $\mathbf{\frac{1}{s_1}\sum_{j=1}^{7} r_j}$, or equivalently:

$$\min z = \sum_{j=1}^{7} r_j. \tag{11}$$

The presented MILP model that assigns collection sites to service days so that the collection sites visited on a given day are as close to each other as possible, consists of minimizing (11) subject to (1)-(10), and will be denoted by ASD ("Assigning Sites to Days").

3 IMPROVEMENT PROCEDURES

Despite the considerable reduction in the amount of variables and constraints of the previous assignment models with respect to other waste collection problems found in the literature, the proposed MILP model was only capable of solving instances of moderate size, very far away from the required 1000 locations of the company. Additionally, the fact that the company wanted to solve this part of the problem in less than five minutes by means of free software calls for the application of some heuristic procedures.

3.1 Reducing the Number of Variables

In our efforts to reduce the complexity of the problem, a first attempt to solve ASD more efficiently was to set variable v to be equal to the total amount of waste to be collected per week divided by the number of service days (the arithmetic mean), that is, $v = 7(\sum_{i=1}^{I}\sum_{k=1}^{2} n_i^k g_k)/s_1$. This way ASD has one variable less, and therefore a reduction in computational time was hoped for. The first we noted is that, despite the intuitive idea that both problems should be equivalent (v free and v equal to the mean), they are not. Consider the following example.

Example 2. *Consider a WC instance in which the total amount of waste to be collected is 4, over 3 service days. Assume that the amounts collected per day are $w_1 = w_2 = 1, w_3 = 2$. Take $\varepsilon = 1/3$. It is easy to see that, for $v = 3/2$, $w_j \in [(1-\varepsilon)v, (1+\varepsilon)v] = [1,2]$. Nevertheless, if we fix $v = 4/3$, we have that $[(1-\varepsilon)v, (1+\varepsilon)v] = [8/9, 16/9]$ and therefore w_3 falls out of the allowed interval. Therefore, with $\varepsilon = 1/3$, the previous instance would be infeasible if v is fixed to be the mean, and it would be feasible if v is free.*

In order to see whether or not the difference between the performance of both models (v free and v fixed) is significant, we tested both models over 12 different-sized instances. The values of I and the objective values obtained with CPLEX after 5 minutes for each instance and each model are shown in Table 1. The values of s_k, f_k, c_k, g_k are the same as in Example 1.

Table 1: Tests to fix variable v.

I	16	25	36	49	64	81
v free	9.5	15	17.5	22.5	29.5	35
$v =$ mean	9.5	13	18	24.5	31	35
I	100	121	144	169	196	225
v free	47	50	61	68	73	83.5
$v =$ mean	45.5	49	60	69.5	75	81

From these data, a hypothesis test with 95% confidence interval does not reject that the objective values obtained with both models are equal (p-value = 0.93). In other words, fixing the value of v does not seem to improve the objective function value with the allowed computation time. This, together with Example 2, made us keep the ASD formulation described before without fixing variable v.

3.2 Clustering

Solving a 1000 location problem is a real challenge. However, a practical observation brings an important venue for problem size reduction. Often, a relatively short street has two container locations, spaced less than a few meters away. Additionally, and specially in road crossways, containers are located in the corners of several roads. All these different locations can be easily clustered into a single location with very little expected degradation in the final solutions. Such scenarios are very likely in large cities which have thousands of containers but these are always tightly placed in streets. By following simple clustering approaches (like K-nearest neighbor) we can reduce the size of the original problem by a factor of K in polynomial time.

These simple clustering approaches were fast and easy to implement but we applied a different approach. Instead of applying a K-nearest neighbor we employ effective Traveling Salesman Problem (TSP) heuristics. The Lin-Kernighan heuristic (Lin and Kernighan, 1973) is one of the most well known and powerful heuristics. More specifically, an effective implementation (Helsgaun, 2000) is employed. This Helsgaun heuristic is regarded as the most effective method for solving the TSP which can yield close to optimal solutions in very short computational times with problems with hundreds or even thousands of nodes. The procedure is the following: We construct a TSP problem with all the locations and run the Helsgaun heuristic. The result is the shortest TSP tour that traverses all locations, starting and finishing at a given one. Then we apply a K clustering procedure. We start from the first location in the tour and create a new virtual location with this one and the following $K-1$ locations. Location $K+1$ is clustered with lo-

Table 2: Instances tested and their characteristics.

Instance	Locations (I)	s_1	f_1	s_2	f_2	g_1	c_1	g_2	c_2	Feasible time tables	ε value
1	40	6	2	6	2	10	45	5	25	7	0.2
2	40	6	3	6	2	10	35	5	25	35	0.05
3	70	6	2	6	2	10	45	5	25	7	0.2
4	70	6	3	6	2	10	35	5	25	35	0.05
5	100	6	2	6	2	10	45	5	25	7	0.2
6	100	6	3	6	2	10	35	5	25	35	0.05
7	130	6	2	6	2	10	45	5	25	7	0.2
8	130	6	3	6	2	10	35	5	25	35	0.05
9	260	6	2	6	2	10	45	5	25	7	0.2
10	260	6	3	6	2	10	35	5	25	35	0.05
11	520	6	2	6	2	10	45	5	25	7	0.2
12	520	6	3	6	2	10	35	5	25	35	0.05

cations $K+2,\dots,2K$ and so on. This procedure continues until the last location of the tour is considered. The result is a I/K location problem in which each virtual location contains K original locations.

We expect the deterioration in the solution to be very small. Note that after applying the Helsgaun heuristic, the locations at each clustered virtual location are most of the time very close to each other. Note that many other clustering methods and algorithms are possible.

4 INITIAL COMPUTATIONAL RESULTS

In the following we present the results of the simple ASD method when run with real instances. Some results with the application of the clustering procedure will be given as well.

We employ a set of 12 real instances with data obtained from the company. These range from 40 to 520 locations in size. While still deemed as "medium", they serve as a good benchmark of what can be done with the ASD method alone. All instances contain two fractions, whose full parameters are given in Table 2.

As we can see, the mix of configurations yield two cases with either 7 possible timetables per location or 35. Note that in the largest case of 520 locations and 35 possible timetables, the total number of x_i^ℓ variables is $520\times 35 = 18,200$ which is a really big number of binary variables considering the maximum allowed computational time (5 minutes) with free software (in all tested instances this time limit is reached, i.e., no solution found is guaranteed to be optimal).

All tests are carried out in a high performance computing cluster with 30 blades, each one containing 16 GBytes of RAM memory and two Intel XEON E5420 processors running at 2.5 GHz. Note that each processor has 4 physical computing cores (8 per blade). There is no parallel computing. The 30 blade servers are just used in order to divide the workload and experiments. Experiments are carried out in virtualized Windows XP machines, each one with one virtualized processor and 2 GB of RAM memory. We solve all 8 instances first using IBM ILOG CPLEX 12.2 with a single thread with a maximum CPU time limit of 5 minutes. Later, results in the same conditions with the freely available 2.7 version of COIN-OR Branch-and-cut MIP solver, also referred to as CBC (https://projects.coin-or.org/Cbc) will be given. We also employed LP-Solve 5.5 (http://sourceforge.net/projects/lpsolve/). However, this free solver has a much inferior performance and in these initial experiments, results are reported with just CPLEX 12.2 and CBC 2.7.

We show the final results using CPLEX after the assignment carried out by ASD both in total radii and radii per day of the week (both measured as the Euclidean distance between latitude-longitude like coordinates) in Table 3 as well as in total collected waste (in kilograms) in Table 4.

The results with CBC are presented in Tables 5 and 6.

It is observed that CPLEX provides smaller radii than CBC for all instances except 6 and 8.

As regards the total collected waste we can see that it is fairly constant throughout the days for all instances. The observed fluctuations are often seen at days before and/or after the "break". Note that all instances have 6 service days in the week and therefore, after the day without service, a larger amount of waste needs to be collected. When comparing CPLEX and CBC we have added a column with the standard deviation of the collected waste through the days. It appears that CBC has the upper hand. Although the small number of tested instances precludes us from drawing strong conclusions, it seems that CPLEX is not performing alarmingly better than CBC. Therefore, our objective of employing a freely available

Table 3: CPLEX Radii results for the instances (*no feasible solution obtained for instance 12 in 5 minutes, the reported solution was obtained after 14 minutes and 37 seconds).

	Radii							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Radii sum
1	69.5	96.5	31.5		69.5	96.5	28.5	392
2	91		89	84.5	89	80	96,5	530
3	82.5	88	95	82.5		88	88.5	524.5
4	77	162	109	179.5	78		179.5	785
5	68		66.5	179.5	57	63	179.5	613.5
6	150		179.5	126	174.5	139	175.5	944.5
7	69	70.5	202		78	112	177	708.5
8		158.5	196	172	158.5	186	202	1073
9	101.5		294	187.5	90.5	241.5	240.5	1155.5
10	286.5		268.5	286.5	285.5	250	278.5	1656
11		274	294	148.5	244	294	155.5	1410.5
12*	280	298	285.5	298.5		298.5	276	1736.5

Table 4: CPLEX total collected waste at each day for the instances (*no feasible solution obtained for instance 12 in 5 minutes, the reported solution was obtained after 14 minutes and 37 seconds).

	Total collected waste							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	St. Dev
1	1560	1605	1545		2080	2275	1820	304.59
2	1720		1875	1875	1885	1760	1770	72.35
3	3760	2670	2655	2820		3965	2820	586.89
4	3065	3240	3015	3010	3050		3310	127.59
5	3625		5385	5259	3740	3600	4290	819.10
6	4200		4575	4490	4250	4215	4205	165.79
7	4275	4260	6375		6150	6320	5940	1007.50
8		5770	5740	5280	5305	5410	5815	274.65
9	7245		9590	8700	6460	6405	6575	1342.33
10	7865		7875	7855	7125	7125	7130	404.46
11	19315		19135	12900	12900	15360	12895	3100.07
12*	14795	14790	15035	16315		16320	15250	717.72

Table 5: CBC Radii results for the instances.

	Radii							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Radii sum
1	83.5	42	89		83.5	50.5	100.5	449
2	112.5		103	89	98	139.5	105.5	647.5
3	136.5	158.5	61	136.5		158.5	73.5	724.5
4	174.5	113	110.5	179.5	84.5		179.5	841.5
5	119.5		99.5	179.5	119.5	97	179.5	794.5
6	162		175.5	162	69.5	187	170	926
7	196	148.5	92.5		196	148.5	92.5	874
8		202	107.5	170.5	166	172	194	1012
9	185.3	154.8	294.2	185.3	158.3	294.2		1272.2
10	285.6	256.4	269.4	275.0	294.2	286.5		1667.2
11	229.6	283.9	289.1	229.6	283.9	289.1		1605.1
12	298.3	263.7	290.6	283.9	298.3	280.1		1714.9

solver is satisfied.

We now take the four largest instances: 7-12, and apply the clustering procedure prior to the resolution of the ASD model by CBC. $K = 2$, meaning that the clustering process reduces the original locations by a factor of two. The results are given in Tables 7 and 8.

As can be seen, in almost all cases both the total radii as well as the standard deviation of col-

Table 6: CBC total collected waste at each day for the instances.

	Total collected waste							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	St. Dev
1	2010	2025	2000		1395	1415	2040	317.29
2	1800		1795	1780	1810	1810	1890	38.78
3	3885	3775	2780	2745		2760	2745	555.08
4	3265	3195	2955	3050	2980		3245	137.00
5	4700		4805	4920	3300	3340	4870	780.11
6	4460		4195	4140	4110	4500	4530	194.00
7	7040	6570	5000		4965	4835	4910	982.44
8		5870	5335	5400	5885	5385	5445	253.59
9	8625	8410	8600	6390	6380	6570		1153.69
10	7840	7845	7160	7160	7275	7695		332.98
11	17535	17640	17500	13050	13095	13685		2356.30
12	16010	15980	14595	14495	15535	15890		697.43

Table 7: CBC radii results for the instances 7 - 12 after applying the clustering procedure.

	Radii							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Radii sum
7	151	134	124.5		151	124	124.5	809
8		188.5	136	187	136.5	188.5	136	972.5
9	120.5	234.6	219.2	120.5	234.6	219.2		1148.57
10	217.9	294.2	161.6	294.2	150.8	294.2		1412.95
11	241.7	243.9	208.6	241.7	243.9	208.6		1388.49
12	271.2	280.1	205.1	298.3	183.3	298.3		1536.41

Table 8: CBC total collected waste at each day for the instances 7 - 12 after applying the clustering procedure.

	Total collected waste							
Instance	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	St. Dev
7	6595	5840	6220		4575	4545	5545	846.83
8		5790	5490	5765	5245	5725	5305	241.26
9	8620	8500	8580	6465	6375	6435		1174.03
10	7940	7225	7305	7485	7235	7785		302.93
11	17520	17700	17640	13140	13275	13230		2413.81
12	15870	16020	14600	15120	14850	16045		638.81

lected waste are significantly smaller once the clustering procedure is applied, i.e., applying clustering first, optimization with CBC second, and declustering third, performs better than directly optimizing with CPLEX or CBC without clustering/declustering. The explanation is the following: in five minutes solving time, larger models have very little time to converge. Once the clustering procedure is applied, the size of the model is largely reduced and the final results are better. This means that far from deteriorating results, the clustering procedure improves performance.

All instances and results were checked by the company and results have exceeded their expectations. Considering that the results are only preliminary, this is a good outcome. We have achieved, using CBC and for instances of up to 130 locations, to form packed groups of locations that satisfy all constraints: the number of service days is not exceeded, all collection frequencies for both waste fractions are observed, radii for each service day are minimized and the amount of waste collected at each day is much more stable than currently observed by the company.

5 FURTHER IMPROVEMENTS: FEASIBLE SOLUTION FIRST, GREEDY LOCAL SEARCH SECOND

The procedure we propose in this section is based on the following idea: Once a feasible solution has been found by means of the previous integer linear programming problem, we later try to improve the objective function by moving the collection day of the sites located furthest away from their day's center to either the previous day or the following day, provided that such a movement does not make the containers in this site to overflow. An interchange with another collection site will be done in such a way that none of the containers overflow, the amount collected each day remains within the allowed interval, and the sum of radii decreases. This is justified by the fact that ASD can find a feasible solution relatively quickly, but the improvements in the objective function are found very slowly. For instance, a 400-collection-site example found a feasible solution in around 1 second, with objective value equal to 127, but after 5 minutes the value of the objective function had decreased eight units only. A 625-collection-site example, found a feasible solution in 11 seconds, with objective function equal to 163.5, and after 5 minutes the improvement in the objective function was also quite marginal.

Therefore, we propose to run the ASD until a feasible solution has been found and then to run a local search procedure. The pseudocode of this procedure is as follows:

For a given day j, let $F_j, \overline{r}_j, \overline{t}_j, \overline{w}_j$ be the collection sites visited, the radius, the center, and the amount of collected waste, respectively. For the sake of notation, we let $u(i, j_1, j_2)$ be the amount of waste collected at site i on day j_2 if its collection day has been moved from j_1 to j_2 and the rest of its schedule remains unchanged. Let e_i be the amount collected at site i if no changes in its schedule have been done.

1. Set $E = \emptyset$. For each j, find $i_j : \max_{i \in F_j} d(i, \overline{t}_j) = d(i_j, \overline{t}_j)$. Let t'_j and r'_j be the center and the radius of the new set $F'_j = F_j \setminus \{i_j\}$. Let $q_j = \overline{r}_j - r'_j$.
2. Let $j^* : q_{j^*} = \max_{j \notin E} q_j$. If i_{j^*} cannot be moved neither to the previous day nor to the following, go to step 3. Otherwise, set $h_1 = h_2 = -1$ and go to step 4.
3. Set $E = E \cup \{j^*\}$. If E is the complete set, STOP. Otherwise, go to Step 2.
4. If i_{j^*} could be moved to the previous day, find a collection site $i_1 \in F_{j^*-1}$ that can be moved to F_{j^*} such that $r^1_{j^*-1} + r^1_{j^*} < \overline{r}_{j^*-1} + \overline{r}_{j^*}$, and $w^1_{j^*} = w_{j^*} - e_{i_{j^*}} + u(i_1, j^* - 1, j^*)$ and $w^1_{j^*-1} = w_{j^*-1} - e_{i_1} + u(i_{j^*}, j^*, j^* - 1)$ (the amount of collected waste) are in the allowed intervals. $r^1_{j^*-1}$ and $t^1_{j^*-1}$ are the radius and the center of $F^1_{j^*-1} = (F_{j^*-1} \setminus \{i_1\}) \cup \{i_{j^*}\}$, $r^1_{j^*}$ and $t^1_{j^*}$ are the radius and the center of $F^1_{j^*} = (F_{j^*} \setminus \{i_{j^*}\}) \cup \{i_1\}$. Set $h_1 = (\overline{r}_{j^*-1} + \overline{r}_{j^*}) - (r^1_{j^*-1} + r^1_{j^*})$.
5. If i_{j^*} could be moved to the next day, find a collection site $i_2 \in F_{j^*+1}$ that can be moved to F_{j^*} such that $r^2_{j^*+1} + r^2_{j^*} < \overline{r}_{j^*+1} + \overline{r}_{j^*}$, and $w^2_{j^*} = w_{j^*} - e_{i_{j^*}} + u(i_2, j^* + 1, j^*)$ and $w^2_{j^*+1} = w_{j^*+1} - e_{i_2} + u(i_{j^*}, j^*, j^* + 1)$ (the amount of collected waste) are in the allowed intervals. $r^2_{j^*+1}$ and $t^2_{j^*+1}$ are the radius and the center of $F^2_{j^*+1} = (F_{j^*+1} \setminus \{i_2\}) \cup \{i_{j^*}\}$, $r^2_{j^*}$ and $t^2_{j^*}$ are the radius and the center of $F^2_{j^*} = (F_{j^*} \setminus \{i_{j^*}\}) \cup \{i_2\}$. Set $h_2 = (\overline{r}_{j^*+1} + \overline{r}_{j^*}) - (r^2_{j^*+1} + r^2_{j^*})$.
6. If $h_1 \geq h_2$, move i_{j^*} to day $j^* - 1$, and i_1 to day j^*. Set $F_{j^*} = F^1_{j^*}$, $F_{j^*-1} = F^1_{j^*-1}$, $\overline{r}_{j^*} = r^1_{j^*}$, $\overline{r}_{j^*-1} = r^1_{j^*-1}$, $\overline{t}_{j^*} = t^1_{j^*}$, $\overline{t}_{j^*-1} = t^1_{j^*-1}$, $\overline{w}_{j^*} = w^1_{j^*}$ and $\overline{w}_{j^*-1} = w^1_{j^*-1}$. Go to Step 1. The objective function has improved h_1.
7. If $h_2 > h_1$, move i_{j^*} to day $j^* + 1$, and i_2 to day j^*. Set $F_{j^*} = F^2_{j^*}$, $F^2_{j^*+1}$, $\overline{r}_{j^*} = r^2_{j^*}$, $\overline{r}_{j^*+1} = r^2_{j^*+1}$, $\overline{t}_{j^*} = t^2_{j^*}$, $\overline{t}_{j^*+1} = t^2_{j^*+1}$, $\overline{w}_{j^*} = w^2_{j^*}$ and $\overline{w}_{j^*+1} = w^2_{j^*+1}$. Go to Step 1. The objective function has improved h_2.

We note that the complexity of this algorithm is given by the calculation of the radii and centers of a set of $O(I)$ points, for at most $O(I)$ times. From the algorithms in (Megiddo, 1983), who showed that the minimum enclosing circle could be found in $O(I)$, we derive that this step can be done in $O(I^2)$, and therefore the complexity of one movement of the previous local search algorithm is $O(I^2)$ as well.

An important aspect of the local search procedure is that it can be safely applied after a de-clustering procedure, i.e., the ASD is applied to a clustered problem. We carry out a de-clustering after an initial feasible solution has been found and later we can apply the local search. In this way, much of the precision lost in the clustering is later regained in the local search stage.

With the proposed local search procedure there are four possible algorithms: Simple ASD (no local search, no clustering), ASD with clustering, ASD until feasibility with local search, and ASD until feasibility with clustering, de-clustering and local search. We are currently testing this proposed local search and other variants and will present the results at the conference.

6 CONCLUSIONS AND FUTURE WORK

In this short paper we have studied a realistic waste collection problem that we have divided into two subproblems. In the first subproblem, the only one studied in this initial research, collection locations are assigned to service days in a complex assignment mixed integer linear programming model. In a foreseen second subproblem, each service day is solved separately as a complex vehicle routing problem. This is needed in order to solve the real problem, which involves the collection of up to a thousand nodes each day during the whole week with a fleet of more than 50 waste collection trucks. Although not fully tested, we have proposed several techniques aimed at speeding up the solution of the first subproblem. More specifically, we have presented some variable reduction techniques, a clustering approach and a local search technique. Initial results using commercial and freely available solvers indicate that the solution for the first subproblem is near and viable within short computational times. Initial assessments by the company also indicate that our results are already better than the manual solutions currently employed.

In the next weeks we intend to test the local search procedure. These results will be compared with the previous ones. The desired goal is to reduce computational time and at the same time to improve the quality of the solutions. We also intend to test more real instances and not just 12. Finally, a full fledged comparison with the current status quo at the company will be provided.

REFERENCES

Angelilli, E. and Speranza, M. G. (2002). The application of a vehicle routing model to a waste-collection problem: two case studies. *Journal of the Operational Resarch Society*, 53(9):944–952.

Baustista, J., Fernández, E., and Pereira, J. (2008). Solving a urban waste collection problem using ant heuristics. *Computers & Operations Research*, 35(9):3020–3033.

Beltrami, E. and Bodin, L. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94.

Benjamin, A. and Beasley, J. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research*, 37(12):2270–2280.

Coene, S., Arnoout, A., and Spieksma, F. (2010). On a periodic vehicle routing problem. *Journal of the Operational Research Society*, 61(1):1719–1728.

Golden, B. L., Assad, A. A., and Wasil, E. A. (2002). *The vehicle routing problem*, chapter 10, pages 245–286. SIAM.

Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research*, 126(1):106–130.

Huang, Y., Baetz, B., Huang, G., and Liu, L. (2002). Violation analysis for solid waste management systems: an interval fuzzy programming approach. *Journal of Environmental Management*, 65(4):431–446.

Kim, B.-I., Kim, S., and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642.

Li, Y., Huang, G., Nie, X., and Nie, S. (2008). A two-stage fuzzy robust integer programming approach for capacity planning of environmental management systems. *European Journal of Operational Research*, 189(2):399–420.

Lin, S. and Kernighan, B. (1973). Effective heuristic algorithm for traveling-salesman problem. *Operations Research*, 21(2):498–516.

Maniezzo, V. (2004). Algorithms for large directed CARP instances: urban solid waste collection operational support. Technical Report UBLCS-2004-16, Department of Computer Science, University of Bologna.

Megiddo, N. (1983). Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. COMPUT*, 12(4):759–776.

Russell, R. A. and Igo, W. (1979). An assignment routing problem. *Networks*, 9(1):1–17.

Teixeira, J., Antunes, A. P., and da Sousa, J. P. (2004). Recyclable waste collection planning—a case study. *European Journal of Operational Resarch*, 158(3):543–554.

Wilson, D. C. (1985). Long-term planning for solid waste management. *Waste Management & Research*, 3(3):203–216.