A HEURISTIC FOR THE SOLUTION OF VEHICLE ROUTING PROBLEMS WITH TIME WINDOWS AND MULTIPLE DUMPING SITES IN WASTE COLLECTION

Dirk Sackmann[1], Richard Hinze[2], Ben Michael, Christoph Krieger, Ece Halifeoglu
University of Applied Sciences Merseburg, Eberhard-Leibnitz-Str. 2, 06217 Merseburg, Germany

ABSTRACT
Municipalities have to collect the waste of households at least once a week. Households place their generated waste, which is stored in either bins or bags, on the designated days in front of their properties where waste collection vehicles can then collect the waste. This process is highly repetitive and performed throughout the year, therefore even a small improvement in waste collection and vehicle routing can lead to significant savings. The waste collection vehicle routing problem with time windows (WCVRPTW) differs from the traditional VRPTW by that the waste collecting vehicles must empty their load at disposal sites.
We present a solution to a real world WCVRPTW in a residential waste collection environment with more than 750 requesters per problem instance. In our application private homes are grouped to street segments and are partitioned into disjoint clusters with known capacity afterwards. During clustering the items with shortest assigning paths from centroids are grouped together. The summation of grouped items should not exceed the capacity of each cluster. All clusters have uniform capacity. In addition we suggest a heuristic that solves the routing problem. Each vehicle can, and typically does, make multiple disposal trips per day. A constructive heuristic that is capable of solving the problem is developed and tested, created to improve waste collection in residential areas.

KEYWORDS: Capacitated Clustering, Waiting Time, Extended Savings Algorithm, Municipal Management.

MSC: 90B06

RESUMEN
Las Municipalidades tiene que recoger la asura de las casas al menos una vez por semana. Los vecinos ponen la basura generada, la que se empaquetan en bins o bolsas, en los días designados, enfrente de sus propiedades donde los carros recolectores los recogen. Este proceso es muy repetitivo y se lleva a cabo a lo largo de todo el año, por ello aun una pequeña mejoría en el problema de recolección de basura conlleva grandes ganancias. El problema de la recolección con el carro de la basura es un problema de ruteo de vehículos con tiempo de ventana (WCVRPTW) que difiere del tradicional VRPTW en que los vehículos deben vaciar su carga en los sitios dispuestos.
Nosotros presentamos una solución de un caso real de WCVRPTW en la recolección de basura en las residencias con más de 750 usuarios por instancia del problema. En nuestra aplicación casas privadas se agrupan por segmentos de calle y se particionan en clústeres disjuntos con una capacidad conocida de antemano. Al conglomerar los ítems con rutas más cortas respecto a los centroides son agrupadas juntas. La suma de los ítems agrupados no puede exceder la capacidad de cada clúster. Todos los clústeres poseen capacidad uniforme. Además nosotros sugerimos una heurística que resuelve el problema de ruteo. Cada vehículo puede, y típicamente lo hace, da múltiples viajes para descargar al

[1] dirk.sackman@hs-merseburg.de
[2] richard.hinze@hs-merseburg.de

día. Se desarrolló y probó una heurística constructiva que es capaz de resolver este problema, para mejorar la recogida de basura en un área residencial.

1. INTRODUCTION

The waste collection business is divided into three major areas: commercial, residential and roll-on roll-off. The residential waste collection generally involves servicing private homes. In this paper, we mainly focus on the daily residential waste collection problem. The vehicles leave the depot at the start of the day and must return there before the end of the day. At the end of the day or in between, the vehicle is unloaded at a waste disposal site or at the depot. There are a limited number of identical vehicles available with given capacity to collect the waste, and each tour is served by one vehicle [7]. Private homes are merged to street segments. Those have given time windows in which they must be visited.
This paper is organized as follows. After we briefly review the waste collection problem in section 2, a cluster algorithm and an extended savings algorithm is presented in section 3. Section 4 shows the experimental results. Section 5 provides the concluding remarks and directions of further research.

2. PROBLEM STATEMENT

In vehicle routing problems, the demand occurs at the nodes of the defined graph. However, waste management problems are typically modeled as Arc Routing Problems (ARPs). In ARPs, the aim is to determine a least cost traversal of defined edges or arcs of a graph, subject to some side constraints. The ARP is a natural way to model waste collection problems in cases where most or all bins along a given street segment must be collected at the same time, and most of the street segments must be traversed by the collection vehicle as is the case in densely populated city areas. The problem more closely resembling the one at hand is the capacitated arc routing problem (CARP) introduced by Golden and Wong [8]. In this problem demand occurs along the arcs, some arcs in the network may not require service (i.e., have no demand along them) and the vehicles have a capacity on the total demand that they can serve. Nevertheless we consider the waste management problem as a node oriented capacitated Vehicle Routing Problem, because the demands of the customers are merged to street segments (requestors) in our real world problem. The problem is known as the (capacitated) Waste Collection Vehicle Routing Problem with Time Windows (WCVRPTW). WCVRPTW differs from the traditional VRPTW [17] by that the waste collecting vehicles must empty their load at disposal sites [4, 9, 15]. Since the VRP itself is well-known to be NP-hard [6], the WCVRPTW as a more detailed and complex version of it is even harder to compute. Thus it cannot be solved by optimal (exact) methods in practice and heuristics and metaheuristics are used for the purpose of computing good solutions in acceptable time [2, 9]. Among the most applied solution methods are constructions and clustering/partitioning heuristics [13, 16] along with tabu search [1, 3, 11], genetic algorithms [14], ant colony optimization [10], local search heuristics [12] and neighborhood heuristics [3, 4]. In our approach a k-means algorithm is used to cluster the demand nodes. Other clustering techniques include metaheuristics like iterated local search or unified hybrid genetic search [18]. The routing is done afterwards with a savings algorithm. Mostly there appear two objectives in the literature: minimize the distance travelled by the collection vehicle and minimize the number of collection vehicles [2].
Our real world problem is determined by the following issues:

- multiple dumping sites,
- one depot,
- a given shift duration,
- a given uniform vehicle capacity and,
- given time windows for the requestors.

The symbols used to describe the problem and the solution algorithm are shown in table 1.
Some of the symbols may need further explanations. The considered problem deals with different bin sizes and four different kinds of waste (paper, other recyclable waste, residual waste and organic waste) leading to R different bin types. The routings of vehicles are determined separately for each type of waste. Two different capacity limits are used to describe the vehicles. C_r is the maximum amount of bins of type r a vehicle can serve per day. This value is used to calculate the number of clusters and to limit the cluster capacity. The second capacity q is the maximum weight of waste a vehicle is able to collect without unloading at a disposal site. This value differs from waste type to waste, since different types of waste have different densities. The demands values d_i at each node are calculated by multiplying the bin volume with the mean density of the waste type.

Table 1: List of symbols.

k	Total number of clusters,
N	Total number of demand nodes,
h	Cluster index / centroid node index,
p	Subtour index,
r	Waste bin index,
$0, 0', 0''$	Depot and dumping site indices,
R	Total number of waste bin types,
n_h	Total number of demand nodes in cluster h,
n_{hp}	Total number of demand nodes in subtour p of cluster h,
i, j	Node indices, $i, j = 1, \dots, n$,
d_i	Demand at node i,
b_{ir}	Number of bins type r at node i
C_r	Maximum number bins of type r, which can be served of a vehicle per day, if just type r bins are loaded,
q	Capacity of the vehicles (maximum weight),
s_{ij}	Saving of connecting the nodes i and j,
w_i	Waiting time at node i,
Δw	Change in waiting time for a subtour,
w_i^{max}	Maximum waiting time at node i,
W^{max}	Maximum total waiting time in one cluster,
t_{ij}	Travel time between node i and j,
t_i^s	Service time at node i,
S_i	Arrival time at node i,
TW_i^S	Start of the time window at node i,
TW_i^E	End of the time window at node i,
x_i	Longitude of node i,
y_i	Latitude of node i.

3. A CLUSTER FIRST – ROUTE SECOND APPROACH

There are two main approaches of determining the numbers of used vehicles and the routing of these vehicles. Beside a route first and cluster second approach in vehicle routing problems, there is also a cluster first route second approach. In the following the amounts of waste at a certain address are called demands. A requester with a certain demand can also be called demand node. A cluster is a subset of demand nodes, which should be served by one vehicle. The route is the sequence of demand nodes within the cluster. The approach suggested in this paper determines the clusters first and the route second. The clusters are calculated first, because of the practical requirements of real world problem, of dividing the waste collection area into cohesive subareas. The demands are already assigned to certain dates. Therefore the calculation of clusters and routes considers serving the demands of a single day. The described computation of vehicle routing needs to be performed for each day separately.

3.1. A CAPACITATED K-MEANS ALGORITHM

The selected capacitated k-means algorithm is based on the approach of Geetha, Poonthalir and Vanathi [7]. It assigns all nodes representing the waste, which needs to be collected, to k clusters. The central points of the clusters are called centroids. Each cluster has a certain capacity C_r, specifying the maximum amount of bins, which can be collected by one vehicle within one day.

The algorithm can be represented by six iterative steps. Step (1) determines the initial number of clusters as well as centroids. The k centroids get the coordinates of the k biggest demand points in step (2). The next step is to find the nearest cluster by calculating the shortest linear distance to the centroids. The demand is assigned to this cluster, if the assignment does not lead to an infringement of the cluster capacity. The cluster with the second shortest distance to the demand is selected, if the first cluster is full and so on. Step (4) checks whether all demands could be assigned to clusters. The number of clusters needs to be increased by one and the algorithm starts again with the second step, if it is not possible to assign all demands. In the other case the centroid coordinates are recalculated, based on the demands in the cluster. The algorithm terminates, if the centroid coordinates do not change anymore after recalculation or if the steps (3) to (5) are performed 100 times.

(1) Calculate the minimum number of clusters $k = \sum_{r=1}^{R} \sum_{i=1}^{n} b_i^r / C^r$
(2) Select the k nodes with highest demands as initial centroids
(3) Assign customers in non-ascending order of their demand size to the nearest cluster centroid, without exceeding the cluster capacity until no assignment is possible anymore.
(4) If (all demands are assigned to clusters): go to step (5), else: set $k \leftarrow k + 1$, and return to (2).
(5) Calculate the new longitude and latitude coordinates of the centroids of each cluster: $x_h = \sum_{i=1}^{n_h} x_i / n_h$ and $y_h = \sum_{i=1}^{n_h} y_i / n_h \;\; \forall\, h = 1, \dots, n_h$
(6) If (cluster centroids did not change compared to the previous iteration or 100 iterations are performed): end, else: return to step (3).

The number of clusters determines the minimum number of vehicles, since at least one vehicle serves one cluster.

3.2. A CAPACITATED SAVINGS ALGORITHM WITH TIME WINDOWS

The savings algorithm basic mechanism is to choose the valid combination of two subtours out of a set of subtours, which lead to the highest saving of travel distance. A subtour is a sequence of demand nodes that does not include all demand nodes of a cluster. The step of connecting two subtours is called a merge. The presence of time windows in the considered problems causes the necessity to perform the algorithm based on time and not only on distance. The time calculations are necessary to check, whether a node is served within a time window. The saving of travel time by connecting two subtours is equivalent to the saving of travel distance. The suggested savings algorithm consists of three phases:

(1) Creating initial subtours,
(2) Merging subtours until the capacity limit of the vehicle is reached,
(3) Connecting the complete subtours via the nearest disposal sites.

The steps (1) and (2) are part of the classical savings algorithm [5]. Step (3) is an extension of the algorithm to fit the requirements of waste collection. All of the steps are performed for each cluster and each day of the schedule. The following section describes the routing for one cluster.

Phase 1

The initial subtours are created for each requester of the cluster separately. The tour starts at the depot, goes on with a certain requester, to serve the node, i.e. collect the waste at the node and going on to the nearest disposal site or depot.

Phase 2

In the next stage the subtours are merged until any additional merge would lead to a violation of the vehicle capacity constraint.

The connection of two subtours in this step is explained by the following example in figure 1 and figure 2. Figure 1 shows two different subtours. Each of them beginning and ending at the depot, which is represented by the solid black nodes 0. The shaded nodes represent two different demand nodes i and j.

Figure 1: Two subtours before merge.

A merge in this step of the algorithm cuts off one depot in each subtour and directly connects the demand nodes, as it can be seen in figure 2.

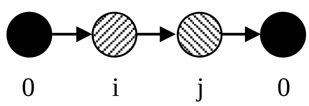


Figure 2: A possible combination of the two subtours.

Considering these pairwise merges of subtours, there are eight possible combinations for any two subtours. An example with two subtours, $p = 1$ and $p = 2$ is presented to illustrate these combinations. The denotations $\overline{1}$ and $\overline{2}$ refer to the reverse tours of the subtours 1 and 2. The possible combinations of these subtours are $1 - 2$, $1 - \overline{2}$, $\overline{1} - 2$, $\overline{1} - \overline{2}$, $2 - 1$, $2 - \overline{1}$, $\overline{2} - 1$, and $\overline{2} - \overline{1}$.
Such a merge is valid if no time windows are violated, the shift duration is not exceeded and the capacity constraint of the vehicle is not exceeded. Additional constraints may be restrictions regarding waiting time. Waiting time occurs if a vehicle arrives at a node before the service time window starts. A detailed explanation of these constraints follows below
The first constraint limits the maximum vehicle load. The loaded quantity of waste collected after the vehicle visited the disposal site or depot the last time ($\sum_{i=1}^{n_{hp}} d_i$) is not allowed to exceed the maximum weight of waste (q).

$$\sum_{i=1}^{n_{hp}} d_i \leq q \qquad \forall p \tag{1}$$

The second constraint ensures that a demand node i is served before the corresponding time window ends. Therefore the sum of arrival time at node i, the waiting time before starting to load the waste and the actual service time for collecting the waste must be lower or equal to the end of the time window.

$$S_i + w_i + t_i^S \leq TW_i^E \qquad \forall i \tag{2}$$

The service at node i needs to start after the start of the corresponding time window. A waiting time w_i is necessary, if the vehicle arrives before the time window starts.

$$TW_i^S \leq S_i + w_i \qquad \forall i \tag{3}$$

The shift duration is assigned to nodes, which do not have an individual time window. The earliest starting time of the shift and the latest end time include all individual time windows. Therefore the shift duration is not violated, if the time window constraints hold.
The algorithm compares the savings of all valid merge combinations and chooses one of the combinations with the highest saving. The starting time of the vehicle can be postponed to avoid waiting times, as long as the time windows of the other nodes are not infringed. The next iteration starts with the then existing new set of subtours. The valid merge combinations are again calculated and compared.
Two possibilities to calculate the saving are presented in this article. The first way of computation is to include the changes in waiting time into the savings calculation. As it can be seen in figure 1 and 2 the travel times from node i to node 0 (t_{i0}), as well as from 0 to j (t_{0j}) are eliminated. The travel time between the node i and j (t_{ij}) needs to be included into the computation of total time of the tour, instead of the two excluded travel times. Also a change of waiting time (Δw) on every node of the new tour influences the saving. The saving increases, if the waiting time is reduced.

$$s_{ij} = t_{i0} + t_{0j} - t_{ij} - \Delta w \qquad \forall i, j \tag{4}$$

The change in waiting time Δw is calculated by equation (5). Δw is negative, if the sum of waiting times ($\sum_{i=1}^{n_{h1'}} w_i$) in the merged tour $n_{h1'}$ is lower than previously occurring total waiting times in the subtours n_{h1} and n_{h2}. It is equal to zero, if both values are the same. An increase in waiting time leads to a positive value of Δw.

$$\Delta w = \sum_{i=1}^{n_{h1'}} w_i - \sum_{i=1}^{n_{h1}} w_i - \sum_{i=1}^{n_{h2}} w_i \tag{5}$$

The second possible specification of savings by using equation (6) does not consider waiting times and just concentrates on travel time.

$$s_{ij} = t_{i0} + t_{0j} - t_{ij} \qquad \forall i \tag{6}$$

Not including waiting times in the savings calculation urges to add additional constraints to prevent extraordinary high waiting times on a single node as well as in total for the whole tour.
The optional restrictions concern waiting times. Each single waiting time at every node is limited to a certain node specific value.

$$w_i \leq w_i^{max} \qquad \forall i \tag{7}$$

The total waiting time in a merged tour is not allowed to exceed a value W^{max}.

$$\sum_{i=1}^{n_{hp}} w_i \leq W^{max} \qquad \forall p \tag{8}$$

The subtours are called complete, if no more merges are possible anymore without violating the capacity constraints. During this step also negative savings would be accepted as long as merges do not violate any of the constraints.

Phase 3

The algorithm enters the third phase, when the subtours are complete. This phase consists of connecting the complete subtours via the closest dumping side. Again the eight combinations mentioned in phase 2 are possible for the merges of any two complete subtours. The time window constraints still need to hold. Additionally the waiting time constraints are required, if the savings are calculated without time windows. The vehicle capacity constraint does not need to be considered anymore since the vehicle is emptied at the connecting dumping site.

The savings in this phase are calculated by equation (9). t_{i0} represents the travel time from node i to depot or dumping site 0. The travel time from the dumping site or depot $0'$ to node j is given by $t_{0'j}$. The index $0''$ stands for the dumping site, which is chosen to connect the two subtours.

$$s_{ij} = t_{i0} + t_{0'j} - t_{i0''} - t_{0''j} - \Delta w \qquad \forall i, j, 0'' \tag{9}$$

Time window constraints may prevent the algorithm from merging all subtours of a cluster into one tour. In such a case additional vehicles are required for the cluster.

4. EVALUATION OF ALGORITHMS

The tests are performed with two different cluster types. The first clusters are real life clusters used by a waste management company before applying the cluster algorithm, called predefined clusters in the following section. The second set of clusters is calculated by the k-means algorithm presented in this paper.

The savings algorithm is tested with configurations: (a) including waiting times in savings (algorithm 1), (b) savings without waiting time (algorithm 2), (c) without time windows (without TW). Each configuration is run on the newly calculated clusters, because the cluster algorithm is evaluated in the first step. The algorithms 1 and 2 deliver the same results for tests without time windows, since there are no waiting times, if time windows do not exist. This problem class is tested, because it may be possible to influence the time windows strategically in order to obtain a better routing. Algorithm 1 and algorithm 2 are compared hierarchically. The first priority is to minimize the number of vehicles used to serve all demands of a day. The second one is to minimize the total time. The total time includes the time spend to collect the waste bins (service time), travel between different nodes, emptying the vehicles at the dumping sites and waiting times.

4.1. TEST INSTANCES

The tests include the data for two types of waste (organic waste, residual waste) of five days in one week. The bins of each waste type have several different sizes in volume. On average about 750 demands including 5130 single bins of residual waste are served each day. That means about 750 nodes are part of the problem every day for this type of waste. These nodes are spread over more than 240 different locations. Some belong to more than one demand, because different bin sizes are segregated into different demand nodes. The mean value for the number of demands of organic waste is approximately 250 per day. Around 150 different locations for this type are visited daily on average.

The service times assumed for the different types of bins are based on experience values and expected to be 20 seconds on average per bin. The time required for emptying the vehicles at dumping sites is 30 minutes each time. The locations of all nodes, i.e. aggregated demand points, the depot and the dumping sites are the real geographic coordinates of the practical case. The travel times are calculated with an average vehicle velocity during travel of 20 km/h. The limit for single waiting time at a node is set to 5 minutes, the maximum total waiting time per vehicle to 1 hour.

4.2. COMPUTATIONAL RESULTS

The first evaluation deals with the capacitated cluster algorithm. The numbers of clusters currently used in the practical case are compared to solutions obtained with k-means algorithm.

Figure 3 shows this comparison for organic waste. The number of clusters created by the algorithm exceeds the number of predefined cluster by two in total. This is caused by the capacity limit within the algorithm, which is set to 100 percent. The existing manually created clusters allow a capacity usage of up to 110-120 percent resulting in a fewer number of clusters necessary to cover the whole demand.

Figure 3: Number of clusters per day for organic waste

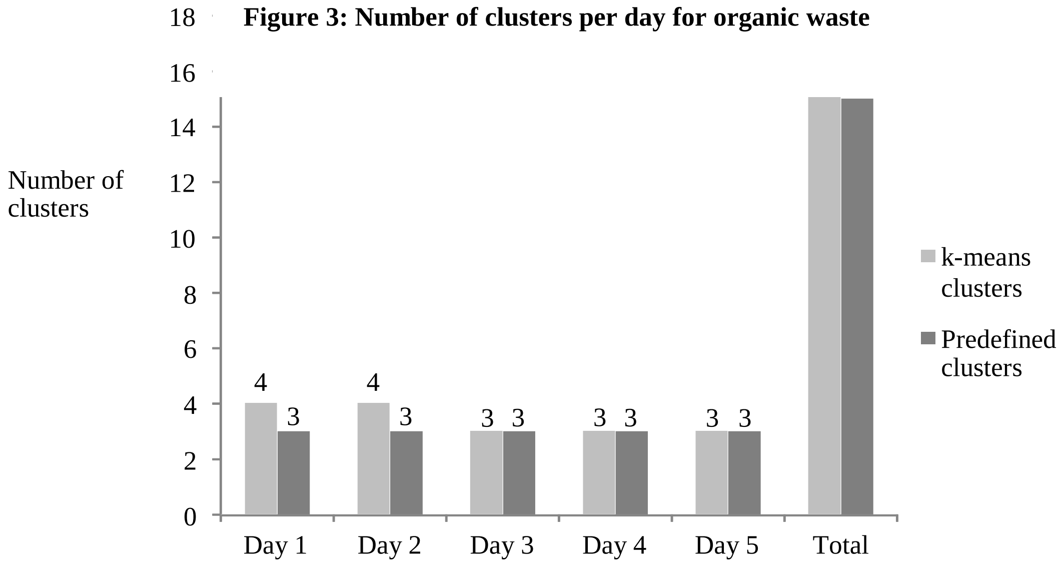


Figure 4 compares the number of manually created cluster with the one obtained with the capacitated k-means algorithm for residual waste. Here the k-means algorithm delivers a smaller number of clusters than there are predefined clusters, despite a higher capacity limit for the manually created clusters. The number of clusters is reduced for two days and increased for just one day by applying the k-means algorithm. The amount of residual waste is higher than the quantity of organic waste, which needs to be collected daily. That is reason why, the number of cluster for residual waste is up to three times higher than the number of clusters for organic waste. The second part of the evaluation considers the total number of vehicles computed by the savings algorithm on the clusters obtained with the k-means algorithm. This comparison is shown in the figures 5 and 6.

Figure 4: Number of clusters per day for residual waste

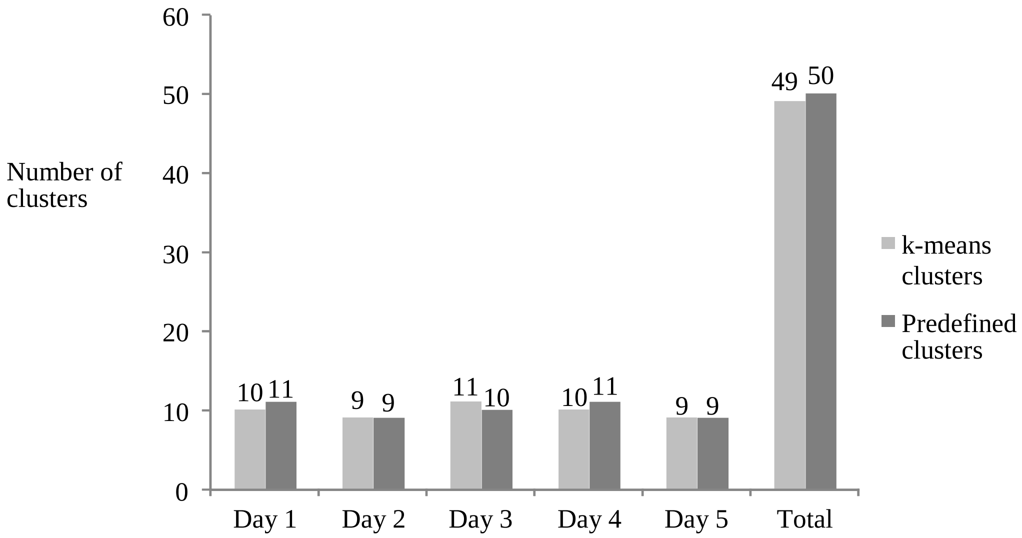
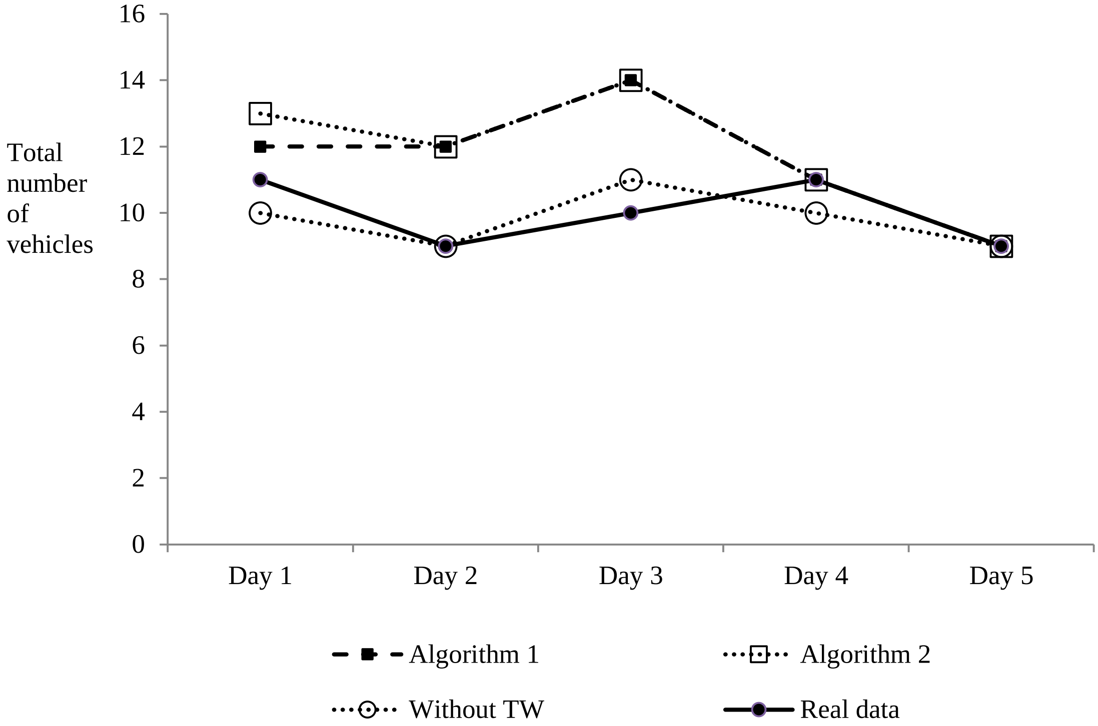
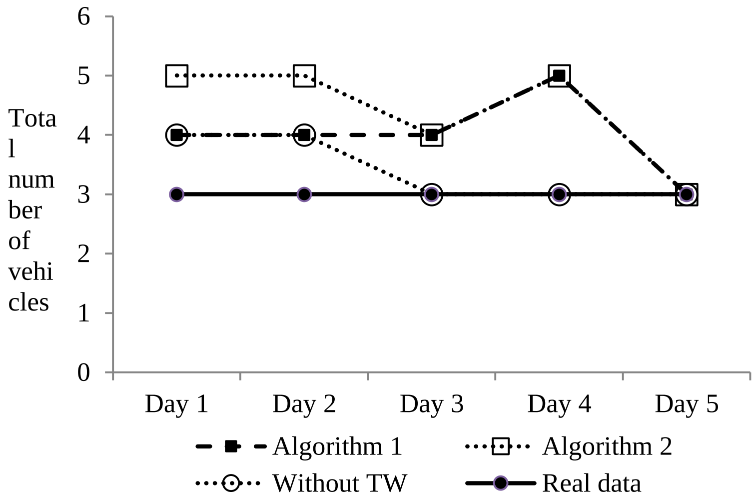


Figure 5: Number of vehicles for residual waste in k-means clusters



The differences between the numbers of used vehicles for organic waste are presented in figure 6. The number of vehicles for the configuration without time windows again equals the number of clusters. Algorithm 2 delivers weaker results for the days 1 and 2, since the solution contains one vehicle more.
The result of the cluster algorithm mainly influences the number of vehicles necessary to serve all demands.

Figure 6: Number of vehicles for organic waste in k-means clusters



Also the use or possible adjustment of time windows influences the results regarding the total number of vehicles obtained by the savings algorithm. Algorithm 1 seems to be more promising than algorithm 2, since it computes in none of ten cases a higher number of vehicles than algorithm 2 does. At the same time it delivers a lower number of vehicles than algorithm 2 does for three cases.

A third evaluation takes the total time into account to get further information on the solution quality of the savings algorithm and its configurations. The results are presented in the tables 2 and 3. Total time is considered to evaluate the algorithms in case of equal numbers of vehicles.

Table 2: Total time [hours] for residual waste in k-means clusters

Day	Algorithm 1	Algorithm 2	Without TW
1	51.63	50.22	51.25
2	53.98	53.98	50.68
3	54.45	55.22	52.29
4	51.21	51.26	51.24
5	47.88	47.88	47.84
Total	259.15	258.56	253.30

Table 3: Total time [hours] for organic waste in k-means clusters

Day	Algorithm 1	Algorithm 2	Without TW
1	19.68	20.13	19.15
2	20.92	20.63	19.85
3	14.45	14.45	14.42
4	15.62	15.62	15.63
5	18.00	18.00	18.00
Total	88.67	88.83	87.05

The total time values obtained by algorithm 1 are higher compared to the solution of algorithm 2, when the routings for residual waste on day 1 and for organic waste on day 2 are calculated. That is caused by the lower number of vehicles for the configuration of considering waiting time changes in the savings. Algorithm 1 obtains lower total time values for two test instances in case of an equal number of vehicles compared to algorithm 2. In total considering the number of vehicles and total times, algorithm 1 delivers better results than algorithm 2 for five of ten instances. The solution quality is equal for the five remaining test instances. Therefore algorithm 1 seems to be the preferred configuration for solving the practical problem.
The fourth column in the tables 2 and 3 shows the total times, if there are no time window constraints except the shift duration. Due to the generally low number of used vehicles in this case, higher total times are observed for several test instances.

5. CONCLUSIONS

In this paper a cluster first route second approach is proposed for the vehicle routing problem with time windows and multiple dumping sites in waste collection. The algorithm consists of a capacitated k-means algorithm and an extended savings. The algorithm tests for five days of waste collection and two types of waste result in cohesive subareas for the collection of waste calculated by the cluster algorithm. The higher the number of requesters, which need to be served, the better is the obtained solutions compared to the existing manually defined clusters. The routing of vehicles computed by the savings algorithm including waiting time delivers better results in the number of vehicles used and regarding total time than the configuration without waiting time. Further research should examine improvement methods for the cluster algorithm, e.g. a lower limit for the waste in every cluster to smooth the capacity usage and to avoid the usage of additional vehicles in the solution of the savings algorithm. An improvement of the savings algorithm may be the combination of savings including waiting time and waiting time constraints, since it showed promising results for a few selected problem instances. Another point is to examine whether additional criteria are useful to decide about a merge of subtours, if the saving of two combinations equals another. Furthermore the algorithm provides the opportunity to examine the influence of changes of time windows on total time and the number of necessary vehicles.
The cluster algorithm already is in practical use. The assignment of demands to clusters is now performed in several seconds instead of several days manual assignment. Future extensions of the planning tool may be a rescheduling option, in case of disruptions during the vehicles are on tour. These disruption events include blocked streets, vehicle breakdowns, deviations in service time or the amount of collected waste.

RECEIVED: JULY, 2016
REVISED: DECEMBER, 2016

REFERENCES

[1] ANGELELLI, E. and SPERANZA, M. (2002): The application of a vehicle routing model to a waste-collection problem: two case studies. In: KLOSE, A., SPERANZA, M. and VAN WASSENHOVE, L. (editors), **Quantitative Approaches to Distribution Logistics and Supply Chain Management**. Berlin, Heidelberg: Springer, 269–286

[2] BELIËN, J., DE BOECK, L. and VAN ACKERE, J. (2012): Municipal solid waste collection and management problems: a literature review, **Transportation Science**, 48, No. 1, 78–102

[3] BENJAMIN, B. 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, No. 12, 2270-2280

[4] BUHRKAL, K., LARSEN, A. and ROPKE, S. (2012): The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context, **Procedia-Social and Behavioral Sciences**, 39, 241-254

[5] CLARKE; G. and WRIGHT J. (1964): Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, 12, No. 4, 568-581

[6] GAREY, M. and JOHNSON, D. (2009): Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: Freeman, 21 print

[7] GEETHA, S., POONTHALIR, G. and VANATHI, P. T. (2009): Improved K-Means Algorithm for Capacitated Clustering Problem, **INFOCOMP Journal of Computer Science**, 8, No. 4, 52-59

[8] GOLDEN, B. L. and WONG, R. T. (1981): Capacitated Arc Routing Problems, **Networks**, 11, No. 3, 305-315

[9] HAN, H. and PONCE-CUETO, E. (2015): Waste Collection Vehicle Routing Problem: Literature Review, **Promet-Traffic & Transportation**, 27, No. 4, 345-358

[10] KARADIMAS, N. and KOUZAS, G. (2005): Urban solid waste collection and routing: The ant colony strategic approach, **International Journal of Simulation: Systems, Science & Technology**, 6, No. 12-13, 45–53

[11] KIM, H., YANG, J. and LEE, K-D. (2009): Vehicle routing in reverse logistics for recycling end-of-life consumer electronic goods in South Korea, **Transportation Research Part D: Transport and Environment**, 14, No. 5, 291–9

[12] MARKOV, I., VARONE, S. and BIERLAIRE, M. (2014): Vehicle Routing for a Complex Waste Collection Problem, **Proceedings of the 14th Swiss Transport Research Conference**, Monte Verit, Ascona, May 14-16 2014

[13] NUORTIO, T., KYTOJOKI, J., NISKA, H. and BRAYSY, O. (2006): Improved route planning and scheduling of waste collection and transport, **Expert Systems with Applications**, 30, No. 2, 223–32

[14] OMBUKI-BERMAN, B., RUNKA, A. and HANSHAR, F. (2007): Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms, **Proceedings of the Third IASTED International Conference on Computational Intelligence**, 91–7

[15] SAHOO, S., KIM, S., KIM, B. and KRAAS, B. (2005): Routing Optimization for Waste Management, **Interfaces**, 35, No. 1, 2005, 24-36

[16] SIMONETTO, E. and BORENSTEIN, D. (2007): A decision support system for the operational planning of solid waste Collection, **Waste Management**, 27, No. 10, 1286–1297

[17] SOLOMON, M.M. (1987): Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, **Operations research**, 35, No. 2, 254-265

[18] VIDAL, T., BATTARRA, M., SUBRAMANIAN, A. and ERDOGAN, G. (2015): Hybrid metaheuristics for the Clustered Vehicle Routing Problem, **Computers & Operations Research**, 58, 87-99

[19] WILLEMSE, E. J., and JOUBERT, J. W. (2012): Constructive Heuristics for the Residential Waste Collection Problem, **Proceedings of the 2011 ORSSA Annual Conference**, 19–28