Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 39 (2012) 241 – 254

Procedia
Social and Behavioral Sciences

The Seventh International Conference on City Logistics

The waste collection vehicle routing problem with time windows in a city logistics context

Katja Buhrkal[a], Allan Larsen[a*], Stefan Ropke[a]

[a]*Department of Transport, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

Collection of waste is an important logistic activity within any city. In this paper we study how to collect waste in an efficient way. We study the Waste Collection Vehicle Routing Problem with Time Window which is concerned with finding cost optimal routes for garbage trucks such that all garbage bins are emptied and the waste is driven to disposal sites while respecting customer time windows and ensuring that drivers are given the breaks that the law requires. We propose an adaptive large neighborhood search algorithm for solving the problem and illustrate the usefulness of the algorithm by showing that the algorithm can improve the objective of a set of instances from the literature as well as for instances provided by a Danish garbage collection company.

Keywords: Waste collection; vehicle routing; reverse logistics; optimisation; case study

1. Introduction

The recent years' intense focus on reduction of emissions has together with the ever-ongoing urbanization of Western world countries led to an increased interest in urban freight transport. According to the European Commission [9] 24% of the goods vehicles which operate in Europe are empty and urban traffic accounts for 40% of the total CO2 emission caused by the transport sector. Thus, a great potential for substantial economic as well as environmental savings lies in reducing urban transport.

The processes for planning, optimizing and controlling logistics and transport activities in urban areas are often referred to as "City Logistics" (see Taniguchi et al. [21]). City logistics can be divided into

[*] Corresponding author. Tel.:+45-45-251502; fax: +45-45-936533.
E-mail address: ala@transport.dtu.dk

 doi:10.1016/j.sbspro.2012.03.105

forward and reverse logistics operations dealing with the flow of goods from the producers to the consumers and the flow from the consumers to recycling or disposal facilities, respectively. Thus, the collection of waste is a central component in the logistic set-up of a large city.

The waste collection problem consists of routing vehicles to collect customers waste within given time window while minimizing travel cost. This problem is known as the Waste Collection Vehicle Routing Problem with Time Windows (WCVRPTW). WCVRPTW differs from the traditional VRPTW by that the waste collecting vehicles must empty their load at disposal sites. The vehicles must be empty when returning to the depot. Multiple trips to disposal sites are allowed for the vehicles. The problem is illustrated in Fig. 1 for a single vehicle and multiple disposal sites.

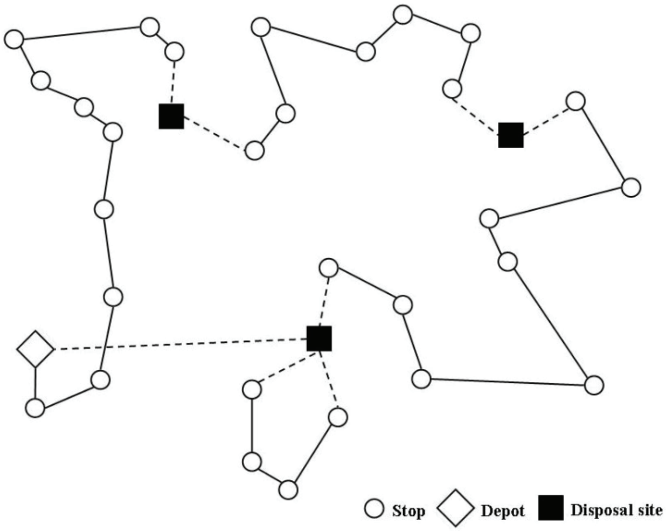

Fig. 1. A route sequence of one vehicle considering disposal operations with multiple disposal sites

1.1. Structure

The structure of the present paper is the following: Section 2 discusses the literature which has previously studied WCVRPTW or other similar and relevant problems. The WCVRPTW is then formulated formally and modeled in section 3. The subsequent section 4 deals with two case studies on real-life waste collection problems. The two cases have different additional constraints which are modeled. Section 5 presents the selected solution method ALNS and discusses the problem specific modifications which has to be made. Section 6 discusses the results obtained. Finally, our concluding remarks are given in section 7.

2. Literature

The waste collection VRPTW has received some attention in recent years. Kim et al. [11] address a real life waste collection VRPTW with consideration of multiple disposal trips and drivers' lunch breaks. They address the problem by using an extension of Solomon's well-known insertion approach (Solomon, [20]). Ombuki-Berman et al. [15] address the same problem by using a multi-objective genetic algorithm on a set of benchmark data from real-world problems obtained by Kim et al. [11]. Benjamin and Beasley [5] improve the results when minimizing travel distance using a tabu search and variable neighborhood search and a combination of these. A very similar problem, with only one disposal site, is addressed by Tung & Pinnoi [23], where they modify Solomon's insertion algorithm and apply it to a waste collection problem in Hanoi, Vietnam.

Teixeira et al. [22] apply a heuristic approach for a Periodic Vehicle Routing Problem (PVRP) for the separate collection of three types of waste: glass, paper, and plastic/metal. The approach has three phases: define a zone for each vehicle, define the waste type to collect on each day, and select the sites to visit and sequence them. Angelelli & Speranza [3] study the PVRP with intermediate facilities (PVRP-IF). When a vehicle visits an intermediate facility, its capacity will be renewed. They propose a tabu search algorithm for the problem which they apply for estimating the operating cost of different waste-collection systems. The main difference between their problem and ours is the time window constraints, which must be explicitly considered in our problem. Tabu search algorithms are also proposed by Crevier et al. [8] for the multi-depot vehicle routing problem with inter-depot routes, by Cordeau et al. [7] for the multi-depot PVRP, by Brandão & Mercer [6] for the multi-trip vehicle routing and scheduling problem, and by Alonso et al. [1] for the PVRP with multiple vehicle trips and accessibility restrictions. Alonso et al. refer to their problem as the site-dependent multi-trip PVRP (SDMTPVRP), which is very similar to our problem with the exception of the time window constraints. The time windows are considered by Brandão & Mercer along with different capacities of the vehicles and the drivers' working hours in addition to the other constraints. The vehicle routing problem with multiple trips is studied by Petch & Salhi [16]. Azi et al. [4] use adaptive large neighborhood search to solve a vehicle routing problem with multiple trips.

Nuortio et al. [14] present a guided variable neighborhood thresholding metaheuristic for the problem of optimizing the vehicle routes and schedules for collecting municipal solid waste in Eastern Finland. Solid waste collection is furthermore considered by Li et al. [12] for the City of Porto Alegre, Brazil. Their problem consists of designing daily truck schedules over a set of previously defined collection trips, on which the trucks collect solid waste in fixed routes and empty loads in one of several operational recycling facilities in the system. They use a heuristic approach to solve the problem.

3. Problem formulation

In this section we formally define the WCVRPTW. The problem is defined on a graph $G = (V, A)$, where the set of nodes $V = V^d \cup V^f \cup V^c$ consists of a depot $V^d = \{0\}$, m disposal sites $V^f = \{1, \dots, m\}$, n customers $V^c = \{m+1, \dots, m+n\}$ and the set of arcs is $A = \{(i,j) | i, j \in V, i \neq j\}$. Let $K = \{1, \dots, k\}$ be the set of vehicles and let t_{ij} and c_{ij} be the travel time and cost associated with arc (i,j), respectively. Each node $i \in V$ has an associated service time s_i and time window $[a_i, b_i]$ and we define q_i as the amount picked up at a customer $i \in V^c$. It is assumed that all vehicles have capacity C. The objective of the WCVRPTW is to find a set of routes for the vehicles, minimizing total travel cost and satisfying vehicle capacity, such that all customers are visited exactly once and within their time window.

In order to model the problem the depot is split in a start and an end depot $\{0, 0'\}$. The problem can then be modelled using three types of variables: $x_{ijl} \in \{0,1\}$ is one if and only if vehicle $l \in K$ uses arc $(i,j) \in A$, d_{il} represents the accumulative demand at node $i \in V$ for vehicle $l \in K$ and w_{il} represents the start time of service at node $i \in V$ for vehicle $l \in K$. A mathematical model for the WCVRPTW is:

$$min \sum_{(i,j) \in A} c_{ij} \sum_{l \in K} x_{ijl} \qquad (1)$$

$$\sum_{j \in V} x_{0jl} = 1 \qquad \forall l \in K \qquad (2)$$

$$\sum_{i \in V} x_{i0'l} = 1 \qquad \forall l \in K \qquad (3)$$

$$\sum_{i \in V} \sum_{l \in K} x_{ijl} = 1 \qquad \forall j \in V_c \qquad (4)$$

$$\sum_{i \in V} x_{ijl} = \sum_{i \in V} x_{jil} \qquad \forall j \in V_c \cup V_f, l \in K \qquad (5)$$

$$a_i \leq w_{il} \leq b_i \qquad \forall i \in V, l \in K \qquad (6)$$

$$w_{il} + s_i + t_{ij} \leq w_{jl} + (1 - x_{ijl})M \qquad \forall (i,j) \in A, l \in K \qquad (7)$$

$$\sum_{i \in \{0,0'\}} d_{il} = 0 \qquad \forall l \in K \qquad (8)$$

$$d_{il} + q_i \leq d_{jl} + (1 - x_{ijl})M \qquad \forall i \in V \setminus V_f, j \in V, l \in K \qquad (9)$$

$$d_{il} \leq C \qquad \forall i \in V, l \in K \qquad (10)$$

$$d_{il} \geq 0 \qquad \forall i \in V, l \in K \qquad (11)$$

$$x_{ijl} \in \{0,1\} \qquad \forall (i,j) \in A, l \in K \qquad (12)$$

The objective function (1) minimizes the travel cost under the restriction of the following constraints. All k vehicles must leave (2) and return (3) to the depot. Constraint (4) ensures that all customers are serviced exactly once. Inflow and outflow must be equal except for the depot nodes (5). Time windows and service time are covered by (6) and (7). The vehicles must be empty at the start of the routes and at the end of the routes when they return to the depot (8). Constraints (9) accumulate demand for all nodes except the disposal sites. Vehicle capacity is given by (10). Finally (11) and (12) imposes non-negativity and binary variables.

4. A case study on waste collection

Even though waste collection can in general be modeled as WCVRPTW different requirements may exist in real-life. One case of a company from North America has previously been presented in the literature. In this section a case study of a Danish company is also presented. The driver time and rest legislation varies between these two case studies and therefore requires different lunch break and rest considerations. In this section the two cases are presented and changes to the model formulation are given.

4.1. The North American instances

A real-life case of the WCVRPTW has previously been considered in the literature. In Kim et al. [11] the waste management company Waste Management, Inc. in North America is considered. Aside from the formulation of WCVRPTW given in this paper, three additional constraints are used. 1) A limit S on the number of customers to visit on each route, 2) a limit R on the total amount collected at customers for each route, which is a route capacity, and finally 3) a lunch break of duration s^u starting in the interval $[a^u, b^u]$. Constraint 1) and 2) can easily be added to the model formulation, whereas the lunch break is slightly more complicated. Since no specific lunch location is required, one can assume that the lunch break is taken somewhere between two stops i and j. There is thus no additional travel cost for the lunch break. The lunch break u can now be at three different positions (see figure 2): 1) after servicing node i, 2) between traveling from node i and j or 3) before servicing node j. Note it is normally assumed that service cannot be discontinued.

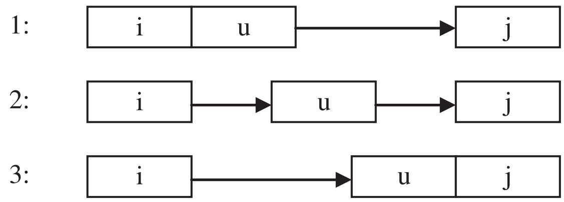

Fig. 2. Possible lunch positions

In order to model the lunch break two new variables are defined. A new binary variable y_{ijl} indicates whether lunch was taken between visiting node i and j by vehicle k. A continuous variable r_l gives the ratio of the driving done between previous and next stop when vehicle l take a lunch break.

$$\sum_{(i,j)\in A} y_{ijl} = 1 \qquad \forall l \in K \qquad (13)$$

$$y_{ijl} \leq x_{ijl} \qquad \forall (i,j) \in A, l \in K \qquad (14)$$

$$w_{il} + s_i + y_{ijl}s^u + t_{ij} \leq w_{jl} + (1 - x_{ijl})M \qquad \forall (i,j) \in A, l \in K \qquad (15)$$

$$a^u + s^u + t_{ij}(1 - r_l) \leq w_{jl} + (1 - y_{ijl})M \qquad \forall (i,j) \in A, l \in K \qquad (16)$$

$$w_{il} + s_i + t_{ij}r_l \leq b^u + (1 - y_{ijl})M \qquad \forall (i,j) \in A, l \in K \qquad (17)$$

$$\sum_{i\in V} \sum_{j\in V^c} x_{ijl} \leq S \qquad \forall l \in K \qquad (18)$$

$$\sum_{i\in V^f} d_{il} \leq R \qquad \forall l \in K \qquad (19)$$

$$0 \leq r_l \leq 1 \qquad \forall l \in K \qquad (20)$$

$$y_{ijl} \in \{0,1\} \qquad \forall i,j \in V, l \in K \qquad (21)$$

Each vehicle must take exactly one lunch break (13). Constraints (14) ensure that a lunch can only be between two nodes i and j if they are connected. The time window constraints (7) are modified to (15), which ensures that the lunch duration is taken into account. Constraints (16) and (17) ensure that the lunch break takes place within its time window. The number of customers serviced is limited in (18) and the route amount in (19). The ratio must be between zero and one (20). Finally the lunch variable must be binary (21).

In the implementation we have chosen to only allow lunch breaks directly after servicing a node. That is $r_l = 0$.

4.2. The Danish instances

The present work is carried out in cooperation with a medium sized waste collection company Henrik Tofteng A/S (HT) located in the Greater Copenhagen area of Denmark. HT serves trade clients and public institutions in the Greater Copenhagen area and has around 50 employees including 35 drivers and 3 dispatchers. The main focus of the company is collection, sorting and disposal of all types of waste except residential waste collection. Secondary, HT performs transport of specific goods for building purposes which require cranes and tippers. HT owns a fleet of 30-35 vehicles of the following types; container trucks, lift trucks, waste collection trucks and a special truck for washing the containers.

In this case study, we are considering the problem involving the waste collection trucks. There are a total of 12 waste collection trucks of which 2 of these are used as reserve vehicles. The waste collection trucks have between 20-50 visits per day per vehicle. Furthermore, the trucks have different capacities which constrain the number of customers they can serve before they have to be emptied at waste disposal sites.

The HT-case also contains certain differences to the WCVRPTW formulation. The vehicles are heterogonous. They may have different start and end locations since drivers often start and end their day at home and only some at the depot. They currently start and end their day at their own choice. They start roughly between 4 and 6 a.m. in the morning with a workday of around 7-9 hours. Some of the vehicles carry keys to specific customer locations that can currently only be serviced by these vehicles.

The company must adhere to the European Union's regulations stipulating the rules for drivers' working and rest hours. The daily driving time must not exceed 9 hours. Since the workday at the company should be less than 9 hours this rule is trivial. At most 4½ hours of driving is allowed before a break of 45 minutes must be taken. The break can be split in two periods of minimum 15 and 30 minutes and the second break should be at least 30 minutes. After the 45 minute break the "clock time" with respect to the 4½ hour limit is restarted. The rules only consider the time spent driving and not the time spent at each stop. Additional rules exist on a weekly and fortnightly basis. Since waste collection consists of multiple stops of about 3-10 minutes the driving time during a day is often between 3-6 hours. Thus, the rules considering longer periods are rarely relevant and we will not consider these. Since the daily driving time might be less than 4½ hours company policy dictate that a 30 minute lunch break should as a minimum be held during the day.

In order to model the HT-case some simplifications have been made. In this paper only homogenous vehicles are considered. Thus, all vehicles start and end at the depot and depot start and end time is used to limit the working day. The driver rest time has been simplified to two parts: 1) a *rest break* always consists of 45 minutes after maximum 4½ hours driving. After the break the counter is restarted. 2) A *lunch break* of minimum 30 minutes must be held once during the day. If a rest break is taken the lunch break is considered covered.

Two new parameters and variables are defined. The duration of a rest break is s^r and binary variable z_{ijl} indicate if a rest break is taken between i and j. The driving time limit is g and h_{il} is a continuous variable for the current driving duration at node i of vehicle k.

$$\sum_{(i,j)\in A} \left(y_{ijl} + z_{ijl}\right) \geq 1 \qquad \forall l \in K \qquad (22)$$

$$y_{ijl} + z_{ijl} \leq x_{ijl} \qquad \forall (i,j) \in A, l \in K \qquad (23)$$

$$w_{il} + s_i + z_{ijl}s^r + y_{ijl}s^u + t_{ij} \leq w_{jl} + \left(1 - x_{ijl}\right)M \qquad \forall (i,j) \in A, l \in K \qquad (24)$$

$$h_{il} + t_{ij} \leq h_{jl} + \left(1 - x_{ijl} + z_{ijl}\right)M \qquad \forall (i,j) \in A, l \in K \qquad (25)$$

$$t_{ij}(1 - r_l) \leq h_{jl} + \left(1 - z_{ijl}\right)M \qquad \forall (i,j) \in A, l \in K \qquad (26)$$

$$h_{jl} \leq g \qquad \forall (i,j) \in A, l \in K \qquad (27)$$

$$h_{il} + t_{ij}r_l \leq g + (1 - z_{ijl})M \qquad \forall (i,j) \in A, l \in K \qquad (28)$$

$$0 \leq r_l \leq 1 \qquad \forall l \in K \qquad (29)$$

$$y_{ijl} \in \{0,1\} \qquad \forall (i,j) \in A, l \in K \qquad (30)$$

$$z_{ijl} \in \{0,1\} \qquad \forall (i,j) \in A, l \in K \qquad (31)$$

Constraints (22) state that all vehicles must take at least one break. A break can only be between two nodes i and j if they are connected (23). The time window constraints (7) are modified to (24), which ensures that both possible break durations are taken into account. The driving duration is maintained by (25) unless a rest break is taken. The starting and ending driving duration after and before a rest break is limited by (26), (27) and (28). The ratio must be between zero and one (29). Finally the break variables must be binary in (30) and (31).

Similar to the North American case, the implementation assumes $r_l = 0$. From a modeling point of view (28) then becomes trivial due to (27). We also assume only one break for each vehicle, i.e. (22) has equality sign. This is due to the company's own policy.

5. Solution methods

The WCVRPTW is a hard problem. It is therefore natural to use a heuristic to solve the problem, as it has been done in (Kim et al., [11]; Ombuki-Berman et al., [15]; Benjamin and Beasley, [5]). We propose to solve the problem using an adaptive large neighborhood search (ALNS) metaheuristic which will be described below, see also (Ropke and Pisinger [17] and [18]). The ALNS heuristic needs an initial solution; this is constructed using the greedy algorithm proposed by Benjamin and Beasley [5] which is described next.

5.1. Benjamin and Beasley greedy heuristic

The Benjamin and Beasley heuristic builds one route at a time and attempts to limit waiting time at customers. It does so by avoiding visiting customers if the visit would incur a waiting time and another customer could be visited without waiting time. The algorithm can be summarized as follows:

1. Open a new vehicle route
2. Insert lunch break if it has not been done yet and the current time is within the lunch break time window.
3. Visit nearest feasible customer that can be serviced without waiting time. Repeat steps 2-3 until no customer found or vehicle near full and close to disposal site*.
4. If vehicle is non-empty visit nearest open feasible disposal site.
5. If possible, advance current time until a customer can be reached within its time window or lunch break can be had (if not done yet), go to step 2. If no more customers can be inserted and non-served customers exist, go to step 1.

*) if the vehicle uses at least 80% of the capacity and the disposal site is closer than the customer considered.

5.2. Adaptive large neighborhood search heuristic

The ALNS heuristic framework was proposed by Ropke and Pisinger [18] and build upon the large neighborhood search (LNS) heuristic proposed by Shaw [19]. Like many other metaheuristics it takes an initial solution as input and attempts to improve upon it through neighborhood search. What sets ALNS and LNS apart from many other metaheuristics is that it searches a very large neighborhood and thereby can make major changes to the current solution in a single step. The neighborhood that ALNS employs is defined implicitly through so called *destroy* and *repair* methods. To get from one solution to the next ALNS first destroys part of the solution and then repairs the partial solution to restore a full solution which most likely will be different from the starting point. For a vehicle routing problem a natural way of destroying a solution is to remove a number of customers from the solution and subsequently let the repair method reintroduce these customers into the solution. This is illustrated in Figure 3a-c. In Figure 4a the current solution with four routes is shown, in Fig. 3b we show the partial solution resulting from the destroy operation that removed 6 customers (large vertexes) and in Figure 3c we show the solution after the repair operation that reinserted the 6 customers.

What sets ALNS and LNS apart is that while LNS uses one destroy and one repair method the ALNS can employ several destroy and repair methods and chooses between them based on their past performance. An ALNS heuristic can be described in pseudo code as follows (x is the current solution, x' is a temporary solution and x^b is the best solution encountered).

1. Input: initial solution x, set of destroy methods Ω^-, set of repair methods Ω^+.
2. $x^b = x$
3. Repeat step a to g until stop-criterion met:
   a. $x' = x$
   b. Choose a destroy and repair method from Ω^- and Ω^+, respectively.
   c. Destroy x'
   d. Repair x'
   e. if x' is accepted then set $x = x'$
   f. if x' is better than x^b then set $x^b = x'$
   g. Update information about performance of destroy and repair methods.
4. Return x^b

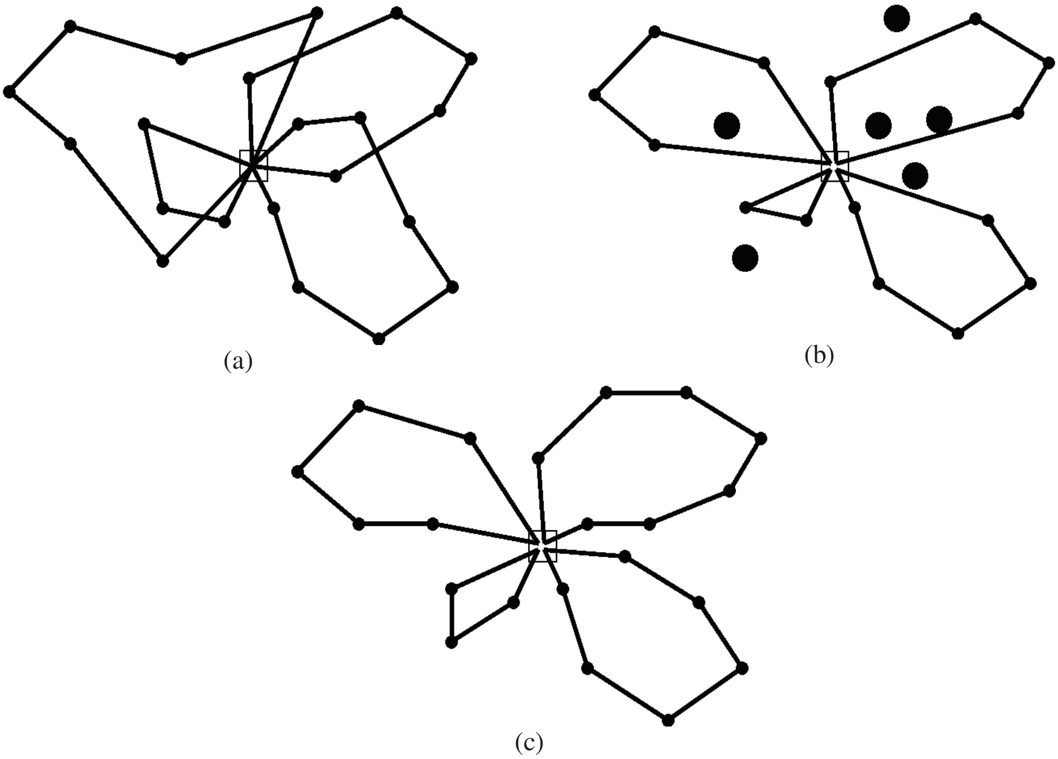

Fig. 3. (a) current solution; (b) partial solution after destroy; (c) new solution after repair

In step 3.e we choose whether or not a temporary solution should be accepted or not. This is done using a simulated annealing acceptance criteria (see e.g. Nikolaev and Jacobson [13]), this means that an improving solution always is accepted and a worse solution is accepted with probability $e^{(f(x')-f(x))/T}$ where $f(.)$ is the objective evaluation and T is the temperature that slowly decreases as the search goes on (meaning that most solutions are accepted early on while only small deteriorations are accepted towards the end of the search). In step 3.b we choose a destroy and repair method to use in the current iteration. This is done using a roulette wheel principle: a destroy method $i \in \Omega^-$ is chosen with probability;

$$p_i = \frac{w_i}{\sum_{j \in \Omega^-} w_j}, \qquad (32)$$

where w_i is a weight associated with method i.

The weights are updated dynamically during the search based on the performance of the method (e.g. when a method contributes to a solution that is accepted the weight is increased). The repair methods are chosen in the same way.

5.3. Destroy and repair methods

In this work we use six destroy methods. The first three are adapted from Ropke and Pisinger [18]:

- *Random remove*. Randomly removes p customers from the solution.
- *Worst remove*. Iteratively removes the worst customer from the solution p times. The worst customer is defined to be the customer whose removal improves the solution the most.
- *Related remove*. Randomly selects one customer for removal. Randomly selects one of the so far removed customers and removes a similar customer. This is repeated until p customers are removed.

These destroy methods only removes customers, not the lunch-break or the disposal visits. While removing the customers we take care that the solution is kept feasibly at all times. The vehicles always start and end at the depot and the lunch break should always be present. The point at which the lunch break and disposal sites are visited may vary by the changes to the customers before and after. However, the number and choice of which disposal sites to visits will remain the same as in the initial solution unless some destroy methods are made to handle this. Thus, three different disposal site destroy methods have been created to allow the heuristic to investigate the entire solution space:

- *Delete disposal.* Randomly selects one disposal for removal. Removes customers directly before and after visit until sufficient load is removed from vehicle to make solution feasible. Note only disposal visits on routes with more than one disposal site can be feasibly removed otherwise the route is closed.
- *Swap disposal*. Randomly selects a disposal visit in solution and an alternative disposal site. Since the disposal site might be further away or have different time windows, customers previous to disposal visit might need to be removed before time windows allows for feasible switch of disposal sites.
- *Insert disposal.* Randomly picks a route and a disposal site and inserts additional disposal visit at the end of route, removing the latest customers to make route feasible.

The following repair moves are used to insert all removed customers into the solution again (further described in Ropke and Pisinger [18]):

- *Greedy insert.* Iteratively inserts the customers that increases the distance the least in the cheapest possible position in the cheapest possible route.
- *Regret-p insert.* Calculates the cost of the best insertion of each customer into each route. If one or more customers can be inserted in less than p routes then let S be the set of customers that can be inserted in fewest routes, otherwise S is the set of all customers. The algorithm inserts the customer from S with the greatest difference in cost between the cheapest route and the p'th cheapest route. Ties are broken by selecting the customer from S with the lowest insertion cost.

Regret-p insert is implemented with $p = \{2,3,k\}$, where k is the number of routes in the solution. If no insertion is possible for a customer it is added to a list of unvisited customers with a penalty of $2M$, where M is the longest distance between two nodes.

5.4. Clustering

For some of the instances tested there exist customers i and j with the same location and time window. It is then cost free to service such two customers directly after each other by the same vehicle. However, due to time windows and capacity it might not be done in the optimal solution. Merging such similar customers into "super customers" does thus not guarantee optimality, but will reduce the number of customers and thus the solution space. Since such customer pairs will most likely be serviced directly after each other by the same vehicle, we have implemented a clustering routine. Customer i and j is replaced by i' if $c_{ij} = 0$, $t_{ij} = 0$, $a_i = a_j$ and $b_i = b_j$. The new customer i' have the information: $c_{i'q} = c_{iq}\ \forall\, q \in V$, $t_{i'q} = t_{iq}\ \forall\, q \in V$, $a_{i'} = a_i$, $b_{i'} = b_i - \min(s_i, s_j)$ and $s_{i'} = s_i + s_j$. The ALNS is tested both with and without clustering.

6. Computational results

In this section we test the ALNS heuristic on two data sets. The first data set was proposed by Kim *et al*. [11]. At http://www.postech.ac.kr/lab/ie/logistics/WCVRPTW_Problem/benchmark.html the instances can be found. The data set contains ten instances ranging from medium sized instances with 99 customers to large scale instances with 2092 customers.

Table 1. Real-life instances provide by Kim *et al* [11]

Size	Benjamin and Beasley (2010)			ALNS results						
(Cust., disposals)	k	Distance	Time	k	Non avg	Non best	Cu.	Cluster avg	Cluster best	Improvement
(99, 2)	3	183.5	2	3	176.03	174.5	83	176.6	174.5	5%
(275, 1)	3	464.5	8	3	455.7	447.6	265	456.4	450.7	4%
(330, 4)	6	204.5	10	6	196.49	182.1	317	190.7	182.4	11%
(442, 1)	11	89.1	18	11	78.998	78.3	442	79.2	78.6	12%
(784, 19)	5	725.6	72	5	650.65	604.1	592	647.8	586.2	19%
(1048, 2)	17	2250.5	116	17	2387.7	2325.7	1008	2370.5	2295.2	-2%
(1347, 3)	8	915.1	105	8	891.17	871.9	532	850.9	828.1	10%
(1596, 2)	13	1364.7	252	13	1385.3	1337.5	867	1230.6	1170.2	14%
(1927, 4)	16	1262.8	285	16	1192.2	1162.5	1855	1180.9	1128.7	11%
(2092, 7)	16	1749.0	266	17	1916.8	1818.9	1869	1650.8	1594.2	9%

The ALNS heuristic has been used with the parameters suggested in Ropke and Pisinger [18]. ALNS was coded with C# and run on a 2.67 GHz PC (Intel® Core™ i7) with 8.00 GB memory. In Table 1 we compare the results produced by ALNS with those reported by Benjamin and Beasley [5]. ALNS has been run ten times for each instance both with and without the clustering version of the algorithm, using the same computation time as used by Benjamin and Beasley [5]. The result can be seen in Table 1. The number of vehicles is the same, except for one additional for the largest instance and the third to last for clustered customers. The reduced number of customers after clustering can be seen in the fourth to last column. The average and best results are shown for both non-clustered and clustered customers. ALNS provides in general better results within the same time as Benjamin and Beasley's variable neighborhood and tabu search combination. Clustering customers improves the results for the larger instances. For the smaller instance results are comparable. Finally Table 1 shows average improvements of 9% from the previous know best results to the best result with ALNS.

In addition to the benchmark instances a new real-life case of Tofteng A/S is considered. Since no time criterion is given, ALNS runs for 200 iterations after the last best global solution was found. The case consists of 8 vehicles, 3 disposals and 228 customers. The company's solution to the problem is known. The time and distance information used in the optimization are calculated from a digital road network. Demand and time windows are unknown, but service time is known. All demands are set to one and capacity to the largest capacity on any route, 40. We have created three scenarios with varying time window width. We consider 2, 4 and 8 hour time windows, all centered around the actual visit time.

The first experiment, summarized in Table 2, attempts to improve upon the routes proposed by the company by simply reordering the visits in the routes using the ALNS heuristic. The first column reports the route number, the second the distance driven on that route in the solution provided by the company. The last 6 columns report the ALNS solutions for each of the three time window scenarios. The distance column report the distance obtained by the heuristic and the Δ column shows the improvement over the company's route. The last row summarizes the results. It is clear that ALNS is able to significantly improve upon the company's solution even when keeping the customer-vehicle assignment fixed. Wider time windows provide slightly bigger improvements, especially for route 6-8. The company would benefit on average 8-13% improvement on the routes alone.

Table 2. Real-life instances provided by Tofteng A/S

			ALNS results					
		Company	± 1 hour		± 2 hour		± 4 hour	
Route	# Stops	Distance	Distance	Δ	Distance	Δ	Distance	Δ
1	36	214	199	7%	197	8%	197	8%
2	37	38	30	21%	29	23%	27	28%
3	26	167	151	10%	151	10%	147	12%
4	40	79	73	7%	67	14%	65	18%
5	14	42	38	9%	38	9%	38	9%
6	26	67	64	4%	63	5%	59	12%
7	22	65	59	8%	57	12%	52	20%
8	39	71	68	4%	65	9%	59	17%
Sum		741	682	8%	666	10%	642	13%

In Table 3 we summarize the results obtained when allowing ALNS to solve the entire WCVRPTW on the company's data. Both the company's own solution and the improved routes are used as an initial solution. ALNS is run ten times for each time window scenario and the best and average solution values provided. The reduction from local route optimization to optimizing globally is considerably. There is a small improvement in the solution quality with a better starting solution and wider time windows allows for larger improvements, from approx. 30% to 45%. Overall using ALNS on the complete waste collection problem gives substantial improvement over sub-optimizing the routes. As we are minimizing driven distance similar reductions (in rough numbers) also applies to fuel consumption and CO2 emissions.

Table 3. Real-life instances provide by Tofteng A/S

			± 1 hour		*± 2 hour*		*± 4 hour*	
Starting solution	Route	Start dist	Distance	Δ	Distance	Δ	Distance	Δ
Company	Avg	741	502	32%	466	37%	409	45%
	Best	741	456	39%	451	39%	376	49%
Pre-optimized routes	Avg	682	486	29%	453	34%	417	39%
	Best	682	457	33%	430	37%	382	44%

7. Conclusion

In this paper we consider the Waste Collection VRPTW. Two different real-lift cases are considered. The two cases have different requirements. Especially the lunch and rest break is complicated. A mathematical modelling formulation is given both for the general WCVRPTW and for the two cases.

The WCVRPTW has been solved using the meta-heuristic Adaptive Large Neighbourhood Search. The method shows improved results for benchmark instances. A new real-life case has been tested and considerable improvements were achieved.

Acknowledgements

This work was supported by the Danish Research Council under the Innovation Consortium program and part of the project entitled "Intelligent Freight Transportation Systems".

References

[1] Alonso F, Alvarez MJ, Beasley JE. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society* 2008; 59: 963-976.

[2] Angelelli E, Speranza MG. The application of a vehicle routing model to a waste-collection problem: two case studies. *Journal of the Operational Research Societyi* 2002a; 53: 944-952.

[3] Angelelli E, Speranza MG. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research* 2002b; 137: 233-247.

[4] Azi N, Gendreau M, Portvin J-Y. An adaptive large neighborhood search for a vehicle routing problem with multiple trips. *Cirrelt-2010-08* 2010.

[5] Benjamin AM, Beasley JE. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research* 2010; 37: 2270-2280.

[6] Brandão J, Mercer A. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research* 1997; 100: 180-191.

[7] Cordeau J-F, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 1997; 30: 105-119.

[8] Crevier B, Cordeau J-F, Laporte G. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research* 2007; 176: 756-773.

[9] European Commission. EUROPA Commission takes action to make urban travel greener, better organised and more user-friendly. http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/1379 , 2010. Last accessed October 2010.

[10] Johansson OM. The effect of dynamic scheduling and routing in a solid waste management system. *Waste Management* 2006; 26: 875-885.

[11] Kim BI, Kim S, Sahoo S. Waste collection vehicle routing problem with time windows. *Computers & Operations Research* 2006; 33: 3624–3642.

[12] Li J-Q, Borenstein D, Mirchandani PB. Truck scheduling for solid waste collection in the city of Porto Alegre, Brazil. *Omega* 2008; 36: 1133-1149.

[13] Nikolaev AG, Jacobson SH. Simulated annealing. In: *Handbook of metaheuristics, 2nd edition*. Springer, New York; 2010, p. 1-39.

[14] Nuortio T, Kytöjoki J, Niska H, Bräysy O. Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications* 2006; 30: 223-232.

[15] Ombuki-Berman BM, Runka A, Hanshar FT. Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms. *Brock University* 2007; Technical Report # CS-07-04, 91-97.

[16] Petch RJ, Salhi S. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics* 2004; 133: 69-92.

[17] Pisinger D, Ropke S. Large neighborhood search. In: *Handbook of metaheuristics, 2nd edition*, Springer, New York; 2010, p. 399-419.

[18] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science* 2006; 40: 455-472.

[19] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems. *Principles and practice of constraint programming, Lecture notes in computer science*. Springer, New York; 1998.

[20] Solomon MM. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research* 1987; 35; 254-265.

[21] Taniguchi E, Thompson RG, Yamada T. Modelling city logistics. In: Taniguchi E, Thompson RG, editors. City Logistics I, Institute of Systems Science Research , Kyoto, 1999, p.3-37.

[22] Teixeira J, Antunes AP, De Sousa JP. Recyclable waste collection planning - a case study. *European Journal of Operational Research* 2004; 158: 543-554.

[23] Tung DV, Pinnoi A. Vehicle routing-scheduling for waste collection in Hanoi. *European Journal of Operations Research* 2000, 125: 449-468.