

Improved route planning and scheduling of waste collection and transport

Teemu Nuortio^{a,*}, Jari Kytöjoki^b, Harri Niska^a, Olli Bräysy^b

^aDepartment of Environmental Sciences, University of Kuopio, P.O. Box 1627, FI-70211 Kuopio, Finland

^bAgora Innoroad Laboratory, Agora Center, University of Jyväskylä, P.O. Box 35, FI-40014 Jyväskylä, Finland

Abstract

The collection of waste is a highly visible and important municipal service that involves large expenditures. Waste collection problems are, however, one of the most difficult operational problems to solve. This paper describes the optimization of vehicle routes and schedules for collecting municipal solid waste in Eastern Finland. The solutions are generated by a recently developed guided variable neighborhood thresholding metaheuristic that is adapted to solve real-life waste collection problems. Several implementation approaches to speed up the method and cut down the memory usage are discussed. A case study on the waste collection in two regions of Eastern Finland demonstrates that significant cost reductions can be obtained compared with the current practice.

© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

The collection of municipal solid waste is one of the most difficult operational problems faced by local authorities in any city. In recent years, due to a number of cost, health, and environmental concerns, many municipalities, particularly in industrialized nations, have been forced to assess their solid waste management and examine its cost-effectiveness and environmental impacts, e.g. in terms of designing collection routes. During the past 15 years, there have been numerous technological advances, new developments and mergers and acquisitions in the waste industry. The result is that both private and municipal haulers are giving serious consideration to new technologies such as computerized vehicle routing software.

This paper describes a study of planning vehicle routes for the collection of municipal solid waste in two different regions in Eastern Finland. In the past, the design of the collection routes has been done manually. The real-life waste collection problem under consideration can be described as follows. Waste is located in containers along the streets of a defined road network. The containers must be collected by a fleet of compactor trucks whose capacity

cannot be exceeded. The considered collection areas involve both heavily and thinly populated areas. There are in total approximately 30,000 waste bins in the studied region. Different types of municipal solid waste are collected separately, and furthermore there are separate containers for each waste type. Each vehicle can typically collect refuse from several hundreds of waste bins before going to the waste disposal site to unload. Only one type of waste can be collected simultaneously by each vehicle. There are various bin types and only deep collection containers and pressing containers are collected separately. Waste must be collected during workdays; collection on Saturdays and Sundays is typically forbidden. The vehicle leaves the depot at the start of the day and must return there before the end of the day. At the end of the day, the vehicle is unloaded at the waste disposal site before returning to the depot in case more than 50% of the vehicle capacity is in use. If the 8-h working day length is exceeded, an overtime pay is incurred. A lunch break of 30 min splits the 8-h working day into two equal halves. Thus, two different tours can be operated for each collecting day; one in the morning and one in the afternoon. In addition, the drivers have two 15-min coffee breaks during the workday. There are a limited number of identical vehicles available with capacity of 26 tons to collect the mixed waste in the considered two regions, and each tour is served by one vehicle.

The waste bins and waste disposal site have given time windows in which they must be visited. More precisely,

* Corresponding author.

E-mail address: teemu.nuortio@uku.fi (T. Nuortio).

the waste disposal site is open from Monday 06:00 to Friday 19:00 and all collection activities must be done within 06:00–22:00. The visits to the waste containers are assumed to follow a given cyclic pattern such that the collections are done similarly within different periods. Collection period and container type are defined by the customer, but the actual waste collection is scheduled within the specified collection period by the waste management company. The visits to same waste containers in consecutive days are forbidden if waste must be collected twice per week or less frequently.

Reliable information on the mass and volume of waste in each container is not available. The amount of municipal solid waste is highly variable and the accumulation of waste depend on several factors such as the number of inhabitants sharing a container, GDP per capita, lifestyle, time of the year, etc. Therefore, the considered waste collection problem is stochastic by nature. In this study, the average accumulation rate of waste in each container type is estimated separately using the historical weight and route. For more information on estimating the waste mass and volume, we refer to Nuortio, Niska, Hiltunen, and Kolehmainen (2004). The time to unload a vehicle at the waste disposal site as well as the time to empty each waste bin is estimated based on historical data using regression analysis. Historical data is also used in estimating the travel times between the waste collection points. For more details, see Nuortio et al. (2004). The objective is to schedule the collection activities and to minimize the total distance traveled by the collection vehicles within each day.

The above described problem can be viewed as a Stochastic Periodic Vehicle Routing Problem with Time Windows and a limited number of vehicles (SPVRPTW). The basic Vehicle Routing Problem (VRP) is one of the most widely studied problems in combinatorial optimization. The objective is to route the vehicles (one route per vehicle, starting and ending at the depot), so that all customers are supplied with their demands and the total travel distance is minimized. The Stochastic VRP (SVRP) arises when some of the elements of the problem are stochastic. The stochastic component could, for instance, be uncertain travel times, unknown demands and the existence of the customers. The general solution approach for SVRP is to generate an a priori solution that has the least cost in the expected sense. For more information on SVRP, we refer to Dror, Laporte, and Trudeau (1989); Gendreau, Laporte, and Sèguin (1996); Bräysy, Gendreau, Hasle, and Løkketangen (2004). The periodic VRP (PVRP) can be defined as the problem of finding routes for all days of a given T -day period. It is often called allocation-routing problem because of the need to assign customers to days in addition to assigning customers each day to vehicles. Although, the PVRP has been used in many applications, it has not been extensively studied in related literature. For a recent literature review, see Bräysy et al. (2004). The VRP with Time Windows (VRPTW) has been studied extensively

during the past decade. In VRPTW, some or all customer points have defined time windows in which they must be visited, e.g. 08:00–10:00 in the morning. The time windows can be also soft. In that case, the time windows can be violated against a given penalty cost. For more information on VRPTW, we refer to recent extensive two-part survey paper by Bräysy and Gendreau (2005a,b). In the literature, the size of the vehicle fleet is often part of the optimization problem. However, in practice, a transportation company usually has a given fixed number of vehicles, and the goal is to use them as effectively as possible. This problem is called in the scientific literature as the VRP with limited number of vehicles. So far, there has been only a little research on this topic. The interested reader is referred to Lau, Sim, and Teo (2003).

In the vehicle routing problems discussed above, the demand occurs at the nodes of the defined graph. However, waste collection 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. Compared to more common node routing problems, customers are here modelled as arcs or edges. For excellent surveys on ARPs, see Eiselt, Gendreau, and Laporte (1995a,b), Assad and Golden (1995); Dror (2000). 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. However, here we decided to use the node routing approach. The rationale is that in our case the waste management company must also consider large thinly populated areas. Another argument is that a more detailed modeling is possible with node routing. Through node routing, it is possible to consider each bin separately and provide a more detailed solution. As the waste management company has detailed data on the waste accumulation levels in each bin, exact stop points of the vehicles to collect each bin, and there are several different types of waste with different collection intervals, we considered node routing more appropriate.

As this problem is computationally very hard, and cannot be solved by optimal (exact) methods in practice, heuristics are used for this purpose. The problem is solved by the Guided Variable Neighborhood Thresholding (GVNT) metaheuristic of Kytöjoki, Nuortio, and Bräysy (2004). The obtained computational results were validated by comparing the generated solutions with the current routes of the waste management company. The main contribution of this paper lies in presenting the conceptual solution model and successful real-life application of the GVNT metaheuristic, including a number of implementation guidelines for designing efficient solution methods for very large-scale practical routing problems. It is shown that significant improvements can be obtained. It is expected that

the practical implementation of the proposed model will provide more information for its improvement.

The remainder of this paper is organized as follows. Section 2 gives a brief review of the recent research on applying intelligent systems to waste collection. The proposed solution model and its input requirements are presented in Section 3, whereas the heuristic routing and scheduling methodologies are explained in Section 4. Our test data and computational results are described in Section 5. The main conclusions of the study are provided in Section 6.

2. Related work

The use of computers to solve waste collection problems has been reported by various researchers during the last 15 years.

Ong, Goh, and Poh (1990) develop a heuristic route first-cluster second approach for optimizing the refuse collection in Singapore. Janssens (1993) describes a mathematical model to determine the optimal vehicle fleet size for collecting waste oil in Antwerp in Belgium. The model consists of demand estimation and a collection model that estimates the number of routes and required travel times. Alvarez-Valdez et al. (1993) suggest a strategy where the dead-heading of the vehicles is minimized by allowing some containers to be moved by the workers to the nearest junction.

Kulcar (1996) reports a case study on optimizing solid waste collection in Brussels. A methodology is developed to illustrate how waste transportation costs can be minimized in an urban area. In addition, several means of transportation including transportation by vehicle, rail and canal are evaluated. Eisenstein and Iyer (1997) investigate the scheduling of garbage trucks in the city of Chicago and develop a flexible routing strategy with regard to the number of visits to the waste disposal site. A dynamic solution method based on Markov decision process is presented. Chang, Lu, and Wei (1997) describe a mixed-integer programming model for routing and scheduling of solid waste collection trucks. The model is integrated with a GIS environment and an interactive approach, and a case study in Taiwan is presented. Smith Korfmacher (1997) presents a case study on designing solid waste collection for urban areas in South Africa and discusses a number of strategies for arranging the collection operations.

Bommisetty, Dessouky, and Jacobs (1998) consider the problem of collecting recyclable materials in a large university campus. The problem is modeled as a periodic VRP, and a heuristic two-phase solution method is suggested. Tung and Pinnoi (2000) address the waste collection activities in Hanoi, Vietnam. The underlying real-life vehicle routing and scheduling problem is formulated as a mixed integer program, and a hybrid of standard VRP construction and improvement heuristics is proposed for

its solution. Mourão (2000) used a route first-cluster second approach where a giant tour is generated first, and then decomposed with a lower-bounding method into a set of routes that are feasible with regard to the vehicle capacity. Bodin, Mingozzi, Baldacci, and Ball (2000) study so called rollon–rolloff VRP faced by a sanitation company. In the rollon–rolloff VRP tractors move large trailers between locations and a disposal facility, one at a time. The authors present a mathematical programming model and four heuristic algorithms for the problem. The same problem is studied earlier in De Meulemeester et al. (1997).

Shih and Chang (2001) develop a two-phase approach for routing and scheduling the collection of infectious medical waste from a set of hospitals. In the first phase, a standard VRP is solved by a dynamic programming method while the second phase uses a mixed integer programming method to assign routes to particular days of the week. Golden, Assad, and Wasi (2001) give a short review and analysis of the real-life applications. Baptista, Oliveira, and Zúquete (2002) present a case study on collection of recycling paper in Portugal. The problem is modeled as periodic VRP and a heuristic approach is presented that consists of initial assignment of collection tasks to days, and interchange moves to improve the solution. Minciardi, Paolucci, and Trasforini (2003) describe heuristic strategies to plan routing and scheduling of vehicles for very large-scale solid waste collection that takes place at district level instead of municipal level. A case study at Geneva, Italy is presented. Teixeira, Antunes, and de Sousa (2004) study the planning of real-life vehicle routes for the collection of different types of urban recyclable waste in Portugal. Heuristic techniques were developed to define the geographic collection zones, waste types to be collected on each day and vehicle routes. Koushki, Al-Duaij, and Al-Ghimlas (2004) present a case study to evaluate the efficiency of municipal solid waste collection in Kuwait. Several indicators to measure the effectiveness are proposed and discussed, and a comparative analysis of the collection costs is reported. Amponsah and Salhi (2004) describe a constructive look-ahead heuristic with tailored improvement mechanisms that are specifically designed for collecting garbage in developing countries. Aringhieri, Bruglieri, Malucelli, and Nonato (2004) study the real-life collection and disposal of special waste such as glass, metal and food. The special waste is collected from containers at collection centers instead of each household. Thus, the problem can be modeled as the rollon–rolloff VRP. Standard heuristic construction and improvement procedures as well as lower bounding procedures are presented.

3. Model

In this section, we describe the developed solution model. First, the input requirements of the model are listed.

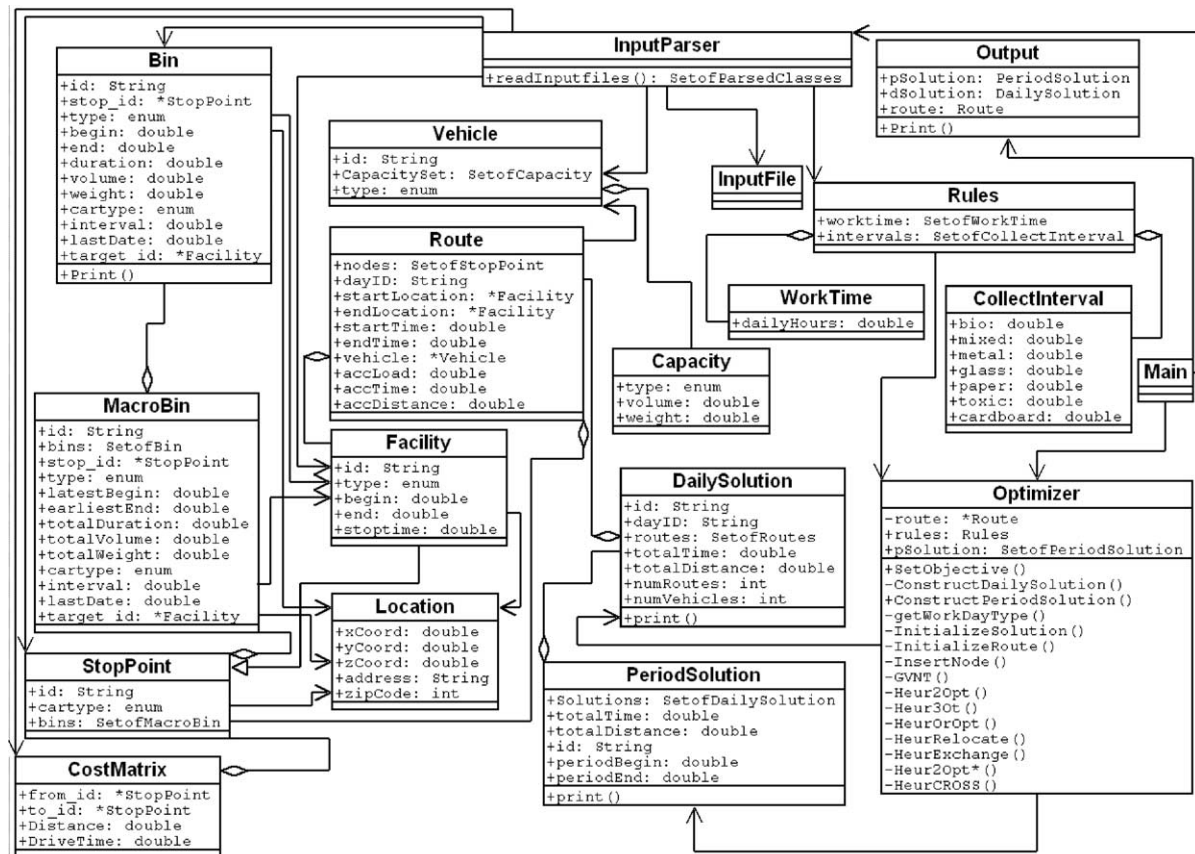


Fig. 1. The conceptual model of the waste collection scheduler and optimizer.

Then, the conceptual model illustrating the considered problem and the used solution approach is presented.

3.1. Input requirements

Inputs for the developed solution method are organized into six data files. These are:

- Bins.** This file contains information on the waste bins. For each bin the ID, location (X- and Y-coordinates), associated stop point ID of the vehicle, waste type, mass and volume of the waste, service time, required time interval for collection, previous collection date and associated waste disposal site are listed.
- Facilities.** This file contains information on the depot and waste disposal sites. For each site, the file lists ID (equals vehicle stop point ID), location (X- and Y-coordinates), facility type (depot or waste disposal site), service time and operating time windows.
- Vehicles.** This file contains information on the available vehicle fleet. For each vehicle, the ID number and volume and weight capacities for different waste types are given.
- Stoppoints.** This file lists all registered stop points for the vehicles based on historical data. A stop point is a location where the vehicle is parked for loading or

unloading. Often several waste bins are collected in the same stop point. For each stop point the file lists the ID number and X- and Y-coordinates.

- Rules.** This file contains information on the allowed maximum daily and weekly working hours, working days, start date of the current planning period and collection intervals.
- Distance matrix.** This file is created by calculating the shortest paths between all pairs of stop points. It describes the distance and drive time for paths connecting the stop points.

3.2. Conceptual model

The used solution approach is based on the conceptual model presented in Fig. 1. As can be seen from the figure, the main function controls the InputParser, Optimizer and Output classes. The InputParser is used to read in the six data files described in Section 3.1. The optimizer is a class of functions described in Section 4 to perform the required scheduling and routing optimization. The user can determine the optimization criteria using the SetObjective function. The default objective is the total distance, but it could be also, e.g. total time or cost. A solution is first initialized and initial daily and period solutions are created

with the `InitializeSolution`, `ConstructDailySolution` and `ConstructPeriodSolution` functions, correspondingly. An attempt is made to improve the daily solutions already during the construction of the first solution, and also after the initial feasible solution has been constructed using the seven well-known improvement heuristics `Heur2Opt`, `Heur3Opt`, `HeurOrOpt`, `HeurRelocate`, `HeurExchange`, `Heur2Opt*` and `HeurCROSS` as described in Sections 4.1 and 4.2. The improvement heuristics are guided by the GVNT metaheuristic of Kytöjoki et al. (2004) described in Section 4.1. The optimization is done according to a number of constraints. Some of these constraints are implicit, and given by the problem, whereas others are set outside, e.g. by work regulations. The latter type of constraints is handled by the `Rules` class. The `Rules` class regulates, e.g. the allowed work time and legislative collection intervals for different types of waste.

As an outcome, the optimizer produces a `PeriodSolution` that gives the allocation of tasks among the days within the collection period (with given `periodBegin` and `periodEnd` dates), i.e. the scheduling solution as well as detailed routes for each collection truck in each day, i.e. the routing solution (`DailySolution`). The `PeriodSolution` consists of a number of `DailySolutions`. The solutions can be evaluated according to total distance or total time of all routes (`numRoutes`). The number of vehicles (`numVehicles`) is typically known in advance, but it can also be an optimization criterion. Each daily solution consists of a number of routes represented by the `Route` class. A route has a given start and end locations as well as associated start and end times, and it consists of an optimally ordered set of stop points, i.e. locations where a vehicle stops to, e.g. collect a waste from a container. The location can be defined according to its *x*, *y*, and *z* (for height) coordinates or according to the physical address and zip code of the location.

A vehicle must be allocated to each route. Each vehicle has its own code for identifying and a given capacity. The capacity is defined separately for each type of waste in terms of both volume and weight. Several capacities may be associated with each vehicle. Each vehicle belongs to one of the predefined vehicle types. The vehicle type is used to model the compatibility of the vehicle with the stop points and waste bins. In other words, a vehicle may or may not be stopped at a given location or used to collect the waste from a given waste bin, e.g. because of the size of the vehicle or required special equipment. The accumulated load (`accLoad`) and the accumulated time (`accTime`) should not exceed the vehicle capacity or violate the work time regulations, respectively. Once the vehicle is full with respect to its capacity, it is driven to the waste disposal site for emptying. The waste disposal site is modeled here with the `Facility` class. The `Facility` class can be used to model also other positions (type) such as the depot where the vehicle is refueled and repaired.

Typically, each route starts and ends at the depot. A visit to a facility has a certain begin and end time as well as

duration (stoptime). The total distance of the daily solution is calculated using the accumulated drive distances (`accDistance`) of each route. The distances and times between each stop point are calculated beforehand as described in Section 4.3 and stored in the `CostMatrix` that is read in with the input parser. Each stop point consists of one or more `MacroBins` and each `MacroBin` consists of one or more waste bins. The `MacroBin` is an abstract class that is used to aggregate waste containers containing same type of waste and having same location and collection interval. These containers are emptied at the same time so it is reasonable to consider them as one to reduce the problem size and facilitate the optimization. Because of different waste types, several `Macrobins` may be associated with each `StopPoint`. Each waste container (`Bin`) has its own identifier (`id`) and it contains only one predefined type of waste. The visits to the container must occur within given time windows, defined by begin and end times. The visits have certain duration, i.e. the time to empty the container that depends e.g. on the size of the container and the distance of the container from the stop point. Each waste container has a specific capacity measured in terms of both volume and weight. There may be restrictions regarding to compatibility of the waste container and the vehicle. These are modeled using the `cartype` attribute that lists the vehicle types that may be used to handle the bin in question. In addition to the collection intervals set by regulations and defined in the `Rules` class, one may set preferred collection intervals separately for each bin using the `interval` attribute. One may also define a target facility (waste disposal site) for the collected waste by specifying the facility identifier. The same attributes apply also to `MacroBins`, although the durations, volumes, weights and time windows must first be aggregated.

4. Methodology

In this section, we describe the applied solution approach. First, the developed routing algorithm, the guided variable neighborhood thresholding of Kytöjoki et al. (2004) is shortly presented. Second, the heuristic scheduling methodology is detailed. Then, the shortest path and travel time calculation are briefly explained. Finally, some implementation guidelines are given.

4.1. Routing algorithm

The routing algorithm of Kytöjoki et al. (2004) consists of two phases. First, a feasible solution is created with a hybrid insertion heuristic. In the second phase, an attempt is made to improve the initial solution with the Guided Variable Neighborhood Thresholding (GVNT) metaheuristic of Kytöjoki et al. (2004). As the name implies, the GVNT is based on three well-known metaheuristic principles: guided local search (Voudouris & Tsang, 1998), variable neighborhood

search (Mladenovic & Hansen, 1997) and threshold accepting (Dueck & Scheurer, 1990), which are applied simultaneously.

The insertion heuristic constructs routes sequentially one at a time until all waste containers have been routed. A route is first initialized with a ‘seed’ container. The seed containers are selected by finding the geographically closest unrouted container in relation to the depot. The remaining unrouted containers are then added into this route according to distance minimization criterion until it is full with respect to the scheduling horizon and/or capacity constraint. If unrouted containers remain, the initializations and insertion procedures are repeated until all waste containers are serviced. Each time after a user-defined number of containers (64) have been inserted in the route currently under construction, an attempt is made to reorder the route with the well-known 2-opt (Flood, 1956), Or-opt (Or, 1976) and 3-opt (Lin, 1965) improvement heuristics. These three operators are applied in turn until no more improvements can be found.

Once the incumbent route is full with respect to the scheduling horizon and/or capacity constraint an attempt is made to improve the current partial solution by applying a set of four well-known inter-route improvement heuristics to all two-route combinations with the just finished route according to the variable neighborhood search strategy of Mladenovic and Hansen (1997). The applied inter-route heuristics are: relocate, exchange (Savelsbergh, 1992), 2-opt* (Potvin & Rousseau, 1995) and CROSS (Taillard, Badeau, Gendreau, Guertin, & Potvin, 1997). If an improvement can be found, the above mentioned three intra-route operators are launched as well. In case all containers have been routed and no more improvements can be found, the second phase is launched. In the second phase, inter-route exchanges are tested with all two-route combinations, and in case of improvements, intra-route operators are also applied. In addition, a guided local search based objective function is used in conjunction with a threshold accepting type limit for the allowed worsening of the objective value. The second phase is repeated until no more improvements can be found. For more details, we refer to Kytöjoki et al. (2004).

4.2. Scheduling algorithm

In this section, we shortly describe some of the basic functionalities of the scheduling algorithm. In this case study, the period length was set to four weeks and most of the containers have 7 or 14 days as their schedule interval, i.e. the containers must be collected once a week or every two weeks.

Before constructing a new daily route, all waste containers are divided in two groups. The first group represents waste containers that require urgent service whereas the less urgent containers are in the second group. The filtering happens through checking the last collection

date. In many cases, the previous collection date is not known, thus all containers must be treated equally. If it is known that a container has been emptied already earlier during the period the next collection time will be defined by the last known collection day and a container specific schedule interval. In the next phase, the containers are sorted in ascending order according to their pickup interval. That is because the containers that require more frequent collection must be assigned first and the rest can be left over for the next day if there is no space for them on the current day.

After all daily routes have been constructed for a specific day, and no more improvements to the daily solution can be found with the GVNT metaheuristic, an attempt to reschedule the containers between the days is made. A move to another day is allowed only if the container must be collected only once during the period and the last collection date is not assigned or known. One must also check that the new collection day is within the collection interval. For example, if the interval is 7 days the first collection must be done within the first 7 days from the start of the period.

The candidates for rescheduling are found by the relocate and exchange inter-route heuristics guided by the GVNT metaheuristic. The other two inter-route heuristics, 2-opt* and CROSS, are not used in the rescheduling context because they would make the implementation too complex and in many cases the improvements would not be possible. Obtaining improvements with the 2-opt* and CROSS would require that collection intervals of the containers in the exchanged route parts are not too strict and none of the containers is collected more than once. After the initial scheduling of the first period is done and all daily solutions have been constructed, the rescheduling between the days is disabled to speed up the route construction and scheduling for the following periods. That is because now the last collection date is known for each container and the collection interval predefines the next collection time.

4.3. Shortest path calculation

The routing and scheduling algorithm requires information on the shortest paths (in terms of travel time or distance) between pairs of waste containers and between waste containers and depot(s) and waste disposal site(s). In this study, the shortest paths are calculated with the well-known Dijkstra algorithm (Dijkstra, 1959) that finds the shortest paths from the ‘source’ node to all other nodes of the defined network. The shortest path source and destination matrix is computed by repeating the application of Dijkstra algorithm iteratively setting each node as source node and calculating the shortest paths to every other node of the network. The distance is used as the only shortest path criterion. That is because distances between the waste containers can be calculated quite reliably from digital road information whereas travel times have a highly stochastic nature. The other reason for using distances is that according

to the haulers the waste collection costs are highly correlated with the total traveling distance.

After calculating the shortest distances between each pair of nodes the traveling times are estimated from the historical GPS-data. Regression analysis is used to estimate traveling speed in each road class, and the time spent on crossroads.

An alternative approach for calculating the shortest paths could be using some Floyd–Warshall based variants. Another improvement could be merging neural network prediction model to shortest path algorithm in order to get better estimates for travel times.

4.4. Implementation details

In this section, we shortly describe some ideas that we used in implementing our solution algorithms to speed-up the computations and to reduce the memory usage at the same time.

One of the most innovative parts of our algorithm is a new and very efficient coding of the distance and time matrices that significantly reduces the memory requirements. Usually, both distance and time between each pair of stop points is represented using separate floating point numbers. Any floating point number mantissa has so called least significant bits. By wisely controlling these least significant bits, it is possible to significantly cut down the memory usage. We were able to present the distance values with just 15–20 bits in the mantissa without causing any bigger impairment of the results, which can be seen here as small noise in the overall computing precision. This way a number of ‘saved’ bits can be used for coding the speed information of the associated shortest path. With standard IEEE float types the computing precision, i.e. the machine epsilon is 1.192×10^{-7} . In case of double types it is 2.220×10^{-16} , and when encoding the speed the overall precision worsens as much as number of bits used for presenting the speed. With this technique both distance and time can be represented with a single floating point number, requiring, e.g. just four bytes memory. Time can be computed in a simple way from the stored distance and encoded speed. To further reduce the memory usage, we calculated so-called twisty-road-curve factors of the actual distances with respect to the Euclidean distances and coded both speeds and curve factors as fixed point numbers using either 16 or 8 bits depending how close the real cost matrix data is to the respective Euclidian values. This makes it possible to calculate actual distances and times based on the Euclidean distances that can be calculated using the x - and y -coordinates. The rationale is that modern processors execute simple calculations faster than they do certain memory operations. Especially, all kinds of cache memory misses should be avoided as much as possible because they generate a lot of penalty clock cycles. As a result, much larger problems can be solved within reasonable amount of memory and computation times are only a bit longer. For more information and methods how to reduce the memory

usage even more with drive time, Euclidean distances and twisty-road-coefficients for huge size cost matrices, we refer to Kytöjoki (2004).

To limit the computational effort in the initial construction heuristic, the waste containers considered for insertion to the current route are limited to the containers with the smallest insertion cost. More precisely, the 1000 smallest insertion costs and the corresponding container for each potential insertion position between two adjacent containers are kept in memory in an ordered one-dimensional table for each container node on the currently constructed route. Thus, containers whose insertion cost is not among the smallest 1000 are ignored. Every time a container is inserted in the route, two ordered lists of 1000 containers and their insertion cost have to be updated (in the beginning an ordered list is initialized for each container) corresponding to the two new potential insertion positions. With this strategy it is possible to quickly determine the cheapest insertion and its cost if we keep a simple heap-priority queue structure.

To speed up the computations of seed containers, all containers are ordered in the beginning primarily according to their distance to the depot and secondarily according to their collection schedule interval using the well-known Quicksort algorithm. In case of equal distance to the depot, the containers with a more frequent collection interval are considered first. The next seed is then determined by finding the first unrouted container in this ordered list.

To further speed up the computations, we use a one-way linked list data-structure in an array to represent the solutions. This structure includes the information of the following container for each container and the array size is simply the number of the containers. In practice, the array just represents the list-next values for each of the containers and any container is found quickly with simple index reference. Using this structure, the changes to the route solution can be done very quickly and in most of the cases in a constant time. For example an addition of a new container u between two adjacent containers i and j is done simply by changing the ‘next-values’ of u to j and i to u . Similarly, one can remove a container or reverse a part of the route very quickly. Compared to the typical one-dimensional table representation of solutions, our strategy is much quicker. For example to add a new container in case of table presentation, one must change the position index of all containers behind the current insertion position, resulting in $O(n)$ complexity.

5. Computational experiments

The suggested solution method has been implemented in Visual C++ 6.0, and it has been tested with a Pentium III 1.0 GHz (512 MB RAM) computer. This case study is based on a real-life waste collection problem in Eastern Finland. Only the collection of mixed waste is included in the study. In this section, we describe the problem data and the computational results.

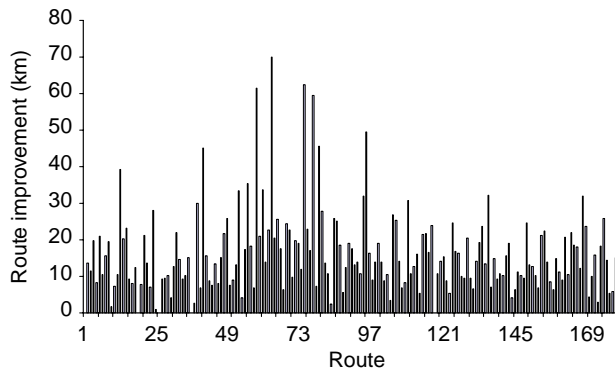


Fig. 2. The difference between current routes and optimized routes in kilometers.

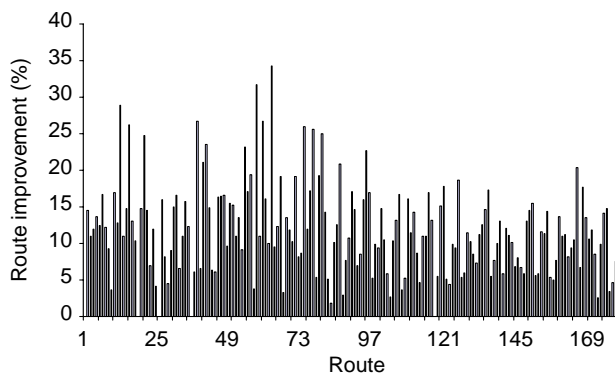


Fig. 3. The difference between current routes and optimized routes in percents.

5.1. Problem data and the experimental setting

The case area consist of the operation area of Jätekuikko Ltd which is responsible for organizing waste management in 18 municipalities (City of Kuopio and the surrounding areas) serving approximately 180,000 people in Eastern Finland. The total area consists of more than 150 collection routes that are mainly handled by two haulers. Two different regions were selected for closer examination in order to re-optimize and schedule the collection routes using the proposed solution methodology. In general, a lot of effort was put on the validation of the historical data in order to provide reliable inputs for the optimization. For more detailed description of the data validation we refer to Nuortio et al. (2004).

The first case region, Pieksämäki, consists of three communes: Pieksämäki, Pieksämäki province and

Virtasalmi. Mixed waste is collected and transported to the transfer station located in Pieksämäki town area from which it is separately transported to the waste disposal site. The waste is mainly collected by a single truck which is daily starting from and ending the journey at the transfer station. First, collection order of single routes was optimized. Second, an attempt to optimize both routing and scheduling was made. The planning period was set to four weeks, due to the fact that a large majority of the waste containers must be emptied at least once in 4 weeks.

The other case region was selected from the Kuopio city area and consists of 82 collection routes in total. In Kuopio city, area population density is much higher than in Pieksämäki area. The purpose was to find an area which is as independent as possible from other collection areas (in the sense that there are not overlapping routes), and two historical time periods in which the most of the containers in that area are emptied. The first period, Kuopio 1, was collected using 6 different vehicles ending up total 58 routes. The other period, Kuopio 2, consists of 61 routes in total. The daily journey starts and ends at the depot located in Southern part of Kuopio city. All waste is finally transported to waste disposal site which is approximate 16 km south-west from Kuopio center.

5.2. The results of single route optimization

The Matlab Matlog toolbox was used to estimate the possible improvements obtained by optimizing the collection order individually for existing single routes, i.e. tackle the traveling salesman problem. The Matlab optimization is based on the well-known Clarke and Wright (1964) savings algorithm and 2-opt improvement heuristic available in the toolbox. All routes are extracted from the Pieksämäki area. The study period (October 2003) consists of 3386 route points (waste containers) and 180 routes. Both remote and dense areas are included. The total distance was used as the optimization criterion. The results are presented in Figs. 2 and 3.

As can be seen from the figures, significant savings can be achieved by optimizing just the collection order of the existing single routes. The difference (in distance) between current route and optimized route varies from 0 to 70 km, averaging to 16 km. The average route improvement in terms of distance is about 12%.

Table 1

Computational results of routing and scheduling optimization for real-life waste collection problems in Eastern Finland

Problem	n	Collection events	Period start	Period end	Estimated total distance (km)	Optimized total distance (km)	Savings (km)	Savings (%)	CPU
Pieksämäki	3225	5709	29.3.04	25.4.04	2693	2587	106	3.9	813
Kuopio 1	4718	14687	6.10.03	2.11.03	5487	3080	2591	43.9	697
Kuopio 2	4353	13738	3.5.04	30.5.04	5483	3073	2410	44.0	1040

5.3. The results of routing and scheduling optimization

The results of routing and scheduling optimization are described in Table 1. The first two columns refer to problem name and size. The next three columns give the period start and end dates and the number of collection events within the period. The estimated total distance, presented in the next column, is defined from the historical route data based on the collection order of the containers. Optimized total distance has been calculated with the model described in Section 3. The last columns show the obtained savings in kilometer and in percents compared with the current practice and the CPU time in seconds.

According to Table 1, significant route improvements are quickly achieved also as a result of combined routing and scheduling optimization. Average saving in Kuopio area is about 2500 km and on the average the routes were improved by nearly 46%. Exceptionally, high improvement performance can be partially explained by the operational limitations. The current operation is strictly bounded to the existing schedules. Thus, by allowing rescheduling it is possible to significantly increase the improvement rate. The used geographical road data has also some inconsistencies and there is not any data available on the shift exchanges. Finally, according to the GPS data on the driven routes several incomplete loads were emptied and in some cases even three dump-visits were done during the workday. In the Pieksämäki area, the data on the visits to disposal site (transfer station) is also missing. The above problems hamper the comparison of the actual and the optimized schedules.

6. Conclusions

In this paper, we have described a case study on a real-life waste collection problem in Eastern Finland. We have presented a conceptual model of the problem that can easily be generalized also to other related problems, and an efficient heuristic solution method based on the guided variable neighborhood thresholding metaheuristic of Kytöjoki et al. (2004). The applied solution methodology integrates efficiently seven well-known improvement heuristics, and the variable neighborhood search, guided local search and threshold accepting metaheuristics as well as a number of new speedup and memory reduction techniques. The experimental results demonstrate that significant savings compared to the current practice can be obtained in both studied levels of optimization: optimization of single routes only, and optimization of both routing and scheduling for the whole collection period.

Acknowledgements

This work was partially supported by the Foundation for Economic Education and Volvo Foundation. This support is

gratefully acknowledged. We would like to thank also the iWaste project financers and partners, especially the Environmental Informatics research unit at the University of Kuopio, National Technology Agency of Finland (Tekes), Jätekuikko Ltd and Ecomond Ltd.

References

- Alvarez-Valdez, R., Benavent, E., Campos, V., Corberan, A., Mota, E., Tamarit, J., & Valls, V. (1993). A computerized system for urban garbage collection. *Topology*, 1, 89–105.
- Amponsah, S.K. & Salhi, S. (2004). The investigation of a class of capacitated arc routing problems: the collection of garbage in developing countries. *Waste Management*, to appear.
- Aringhieri, R., Bruglieri, M., Malucelli, F. & Nonato, M. (2004). A particular vehicle routing problem arising in the collection and disposal of special waste. Presented at Tristan 2004, Guadeloupe, French West Indies.
- Assad, A.A & Golden, B.L. Arc routing methods and applications. In M.O. Ball, T.L. Magnanti, C.L. Monma & G.L. Nemhauser, *Handbooks in operations research and management science 8: network routing* (pp. 375–484). Amsterdam: North-Holland.
- Baptista, S., Oliveira, R. C., & Zúquete, E. (2002). A period vehicle routing case study. *European Journal of Operational Research*, 139, 220–229.
- Bodin, L., Mingozzi, A., Baldacci, R., & Ball, M. (2000). The rollon-rollof vehicle routing problem. *Transportation Science*, 34, 271–288.
- Bommisetty, D., Dessouky, M., & Jacobs, L. (1998). Scheduling collection of recyclable material at Northern Illinois University campus using a two-phase algorithm. *Computers & Industrial Engineering*, 35, 435–438.
- Bräysy, O., Gendreau, M., Hasle, G. & Løkketangen, A. (2004). A survey of heuristics for the vehicle routing problem, part II: demand side extensions. Working paper, SINTEF ICT, Optimization, Norway.
- Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transportation Science*, 39, 104–118.
- Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, part II: metaheuristics. *Transportation Science*, 39, 119–139.
- Chang, N.-B., Lu, H. Y., & Wei, Y. L. (1997). GIS technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering*, 123, 901–910.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, 568–581.
- De Meulemeester, L., Laporte, G., Louveaux, F. V., & Semet, F. (1997). Optimal sequencing of skip collections and deliveries. *Journal of the Operational Research Society*, 48, 57–64.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1, 269–271.
- Dror, M., Laporte, G., & Trudeau, P. (1989). Vehicle routing with stochastic demands: properties and solution frameworks. *Transportation Science*, 23, 166–176.
- Dror, M. (2000). *Arc routing: theory, solutions and applications*. Boston: Kluwer.
- Dueck, G., & Scheurer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90, 161–175.
- Eiselt, H. A., Gendreau, M., & Laporte, G. (1995a). Arc routing problems, part I: the chinese postman problem. *Operations Research*, 43, 231–242.
- Eiselt, H. A., Gendreau, M., & Laporte, G. (1995b). Arc routing problems, part II: the rural postman problem. *Operations Research*, 43, 399–414.
- Eisenstein, D. D., & Iyer, A. V. (1997). Garbage collection in Chicago: a dynamic scheduling model. *Management Science*, 43, 922–933.

- Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4, 61–75.
- Gendreau, M., Laporte, G., & Sèguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, 88, 3–12.
- Golden, B.L., Assad, A.A. & Wasil, E.A. (2001). Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In P. Toth & D. Vigo, The vehicle routing problem (pp. 245–286). Philadelphia: SIAM.
- Janssens, G. K. (1993). Fleet size determination for waste oil collection in the province of Antwerp. *Yugoslav Journal of Operational Research*, 3, 103–113.
- Koushki, P.A., Al-Duaij, U. & Al-Ghimlas, W. (2004). Collection and transportation cost of household solid waste in Kuwait. Waste Management, to appear.
- Kulcar, T. (1996). Optimizing solid waste collection in Brussels. *European Journal of Operational Research*, 90, 71–77.
- Kytöjoki, J., Nuortio T. & Bräysy O. (2004). Two Efficient Metaheuristics for Huge Scale Vehicle Routing Problems. Working paper. Department of Environmental Sciences, University of Kuopio, Finland.
- Kytöjoki, J. (2004). New implementation techniques for search algorithms. Working Paper, University of Jyväskylä, Finland.
- Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*, 148, 559–569.
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44, 2245–2269.
- Minciardi, R., Paolucci, M. & Trasforini, E. (2003). A new procedure to plan routing and scheduling of vehicles for solid waste collection at a metropolitan scale. Presented at Odysseys 2003, Palermo, Italy.
- Mladenovic, N., & Hansen, P. (1997). Variable neighbourhood search. *Computers & Operations Research*, 24, 1097–1100.
- Mourão, M. (2000). Lower-bounding and heuristic methods for collection vehicle routing problem. *European Journal of Operational Research*, 121, 420–434.
- Nuortio, T., Niska, H., Hiltunen, T. & Kolehmainen, M. (2004). Using operational data and neural networks to predict waste formation and collection travel times. Working paper, Environmental Informatics, University of Kuopio, Finland.
- Ong, H. L., Goh, T. N., & Poh, K. L. (1990). A computerized vehicle routing system for refuse collection. *Advances in Engineering Software*, 12, 54–58.
- Or, I. (1976). Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. PhD thesis, Northwestern University, Evanston, USA.
- Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46, 1433(1446).
- Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: minimizing route duration. *INFORMS Journal on Computing*, 4, 146–154.
- Shih, L.-H., & Chang, H.-C. (2001). A routing and scheduling system for infectious waste collection. *Environmental Modeling and Assessment*, 6, 261–269.
- Smith Korfmacher, K. (1997). Solid waste collection systems in developing urban areas of South Africa: an overview and case study. *Waste Management & Research* 15, 477–494.
- Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31, 170–186.
- Teixeira, J., Antunes, A.P. & de Sousa, J.P. (2004). Recyclable waste collection planning- a case study. *European Journal of Operational Research*, to appear.
- Tung, D. V., & Pinnoi, A. (2000). Vehicle routing-scheduling for waste collection in Hanoi. *European Journal of Operational Research*, 125, 449–468.
- Voudouris, C., & Tsang, E. (1998). Guided local search. *European Journal of Operations Research*, 113, 80–119.