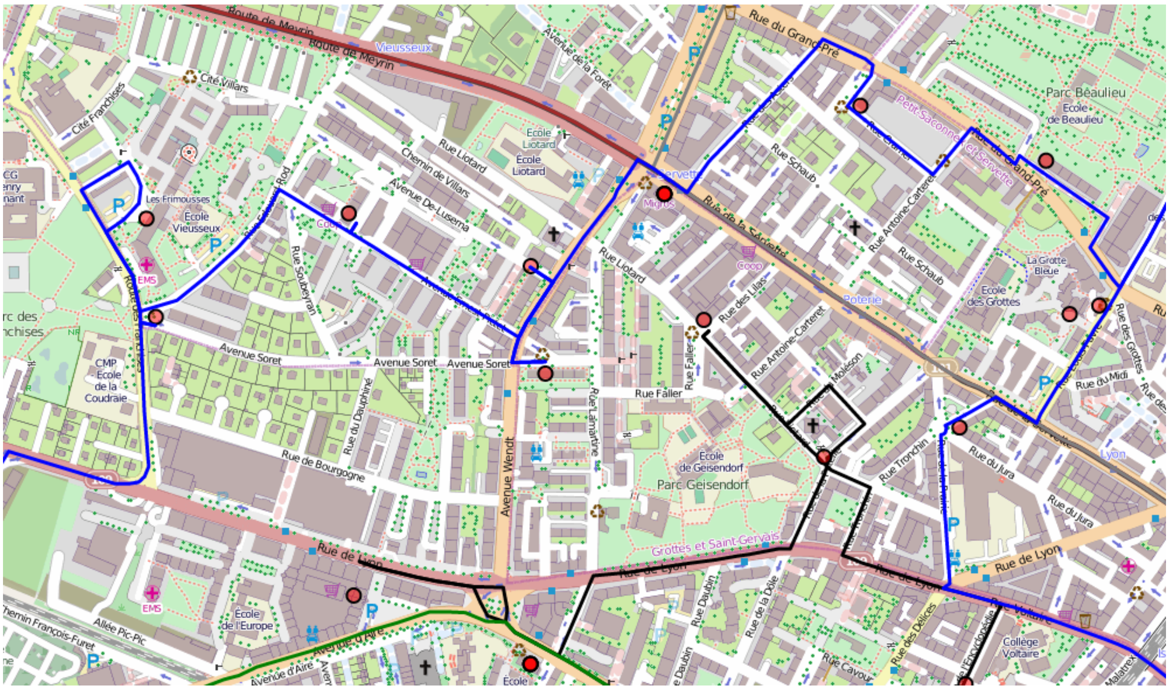


Vehicle Routing for a Complex Waste Collection Problem

Iliya Markov, Ecole Polytechnique Fédérale de Lausanne

Sacha Varone, Haute Ecole de Gestion de Genève

Michel Bierlaire, Ecole Polytechnique Fédérale de Lausanne

April 2014

STRC | **14th Swiss Transport Research Conference**

Monte Verità / Ascona, May 14-16, 2014

Vehicle Routing for a Complex Waste Collection Problem

Iliya Markov
Transport and Mobility Laboratory
School of Architecture, Civil and
Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
phone: +41 (0) 21 693 25 32
fax: +41 (0) 21 693 80 60
iliya.markov@epfl.ch

Sacha Varone
Haute Ecole de Gestion de Genève
University of Applied Sciences Western
Switzerland (HES-SO)
phone: +41 (0) 22 388 17 00
fax: +41 (0) 22 388 17 01
sacha.varone@hesge.ch

Michel Bierlaire
Transport and Mobility Laboratory
School of Architecture, Civil and
Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
phone: +41 (0) 21 693 25 37
fax: +41 (0) 21 693 80 60
michel.bierlaire@epfl.ch

Abstract

We consider a complex waste collection problem, where the residents of a certain region dispose of recyclable waste, which is collected using a fixed heterogeneous fleet of vehicles with different volume and weight capacities, fixed costs, unit distance running costs and hourly driver wage rates. Each tour starts and ends at one of several depots, not necessarily the same, and is a sequence of collections followed by disposals at the available recycling plants, with a mandatory disposal before the end of the tour. There are time windows and a maximum tour duration, which is interrupted by a break after a certain interval of continuous work. Moreover, due to the specificities of different collection regions, there are occasional site dependencies. The problem is modeled as a mixed binary linear program and the formulation is enhanced with several valid inequalities and elimination rules. To solve realistic instances, we develop a local search heuristic, which currently embeds much of the functionality of the mathematical model. The heuristic performs well, as indicated by an optimality gap of 2% compared to the exact solution on small instances. Future work will see improving the model formulation to solve larger instances to optimality and expanding the heuristic to include all of the features of the model.

Keywords

Vehicle routing, Waste collection, Mixed binary linear programming, Local search heuristics

1 Introduction

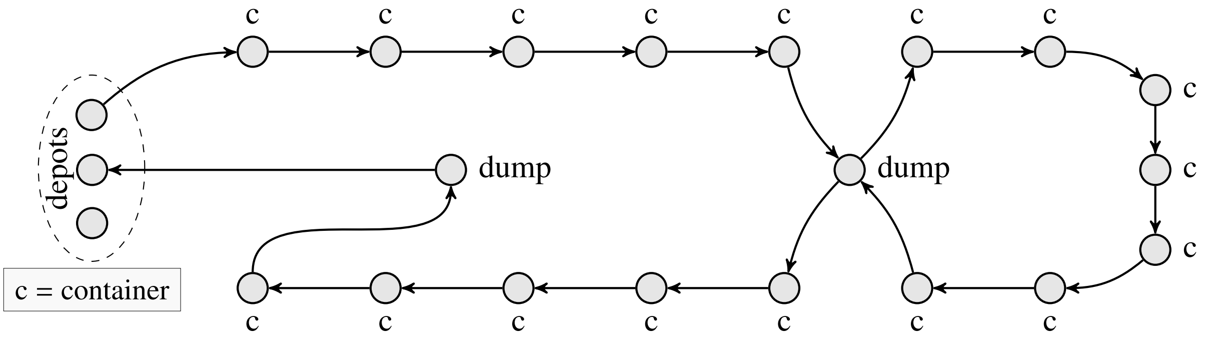
Solid waste collection is one of the most complex logistical problems facing any municipality. In recent years, increased fuel prices and operational costs, environmental and health concerns and the growing regulation burden have caused waste collection companies, both municipal and private, to optimize their collection routes. According to various studies (e.g. Bhat, 1996, Teixeira *et al.,* 2004, Tavares *et al.,* 2009), transportation costs represent between 70 and 80% of all operational costs in waste collection. Therefore even small improvements in the collection routes may lead to substantial savings, which directly affect household fees, municipal budgets and the companies' bottom lines.

This article proposes a solution to a complex waste collection problem, which can be described as follows. The residents of a certain region dispose of recyclable material, such as glass, paper, plastic and aluminum, in designated containers placed at collection points. Each collection point can house up to several containers for a given material type. The waste collection company operates a fixed heterogeneous fleet of vehicles with different volume and weight capacities, fixed costs, unit distance running costs and hourly driver wage rates. A vehicle may be suited for collecting more than one type of material, but only a single type is collected during a tour.

As shown in Figure 1, each tour starts and ends at one of several depots, not necessarily the same, and is a sequence of collections followed by disposals at the available dumps. There is a mandatory visit to a dump just before the end of a tour, i.e. a tour terminates with an empty vehicle. Dumps are recycling plants, each accepting only certain types of material. There could be multiple dumps for each type of material and they can be used when and as necessary during a tour. We consider time windows on dumps and collection points, but not on depots. A tour is limited only by the legal duration of the vehicle driver's working day, which is interrupted by a break after a certain interval of continuous work. Moreover, due to the specificities of different collection regions, we consider site dependencies. Mountainous terrain and narrow streets, for example, are unaccessible with big collector trucks.

The problem is modeled as a mixed binary linear program, which minimizes the number of tours and their spatial and temporal costs. The model introduces several new features to the so-called vehicle routing problem with intermediate facilities (Kim *et al.,* 2006), including a realistic cost-based objective function, multiple depots, a fixed heterogeneous fleet, site dependencies, and a driver break that is not restricted by a time window but depends on when the vehicle started its tour. Moreover, we include a relocation term in the objective function, which incentivizes, rather than enforcing, the vehicle to return to the depot it started from—a situation that mimics the case study we are considering. To solve instances of realistic size, we also develop a local search

Figure 1: Tour illustration



heuristic designed specifically for the problem. Its performance is evaluated by comparison to the optimal solution on small instances of 10 containers and varying time window tightness. Preliminary results show that the heuristic performs favorably with an average optimality gap of less that 2% and a runtime within 1 sec.

The remainder of this article is organized as follows. Section 2 is an overview of related problems from the literature. Section 3 presents the mathematical model formulation. Section 4 describes several classes of valid inequalities and elimination rules for the model and proposes a local search heuristic for solving instances of realistic size. Section 5 offers some preliminary results based on randomly generated instances. Finally, Section 6 concludes and outlines future work directions.

2 Related Literature

In the literature on vehicle routing problems (VRP), the formulation most similar to ours is known as the vehicle routing problem with intermediate facilities (VRP-IF). Beltrami and Bodin (1974) use a simple extension of Clarke and Wright's (1964) savings heuristic to solve a periodic VRP-IF (PVRP-IF) applied to a waste collection problem in New York. Bard *et al.* (1998a) and Bard *et al.* (1998b) consider a distribution context with replenishment facilities, in the latter case integrated in an inventory management framework. Tung and Pinnoi (2000) solve a VRP with only one intermediate facility applied to waste collection in Hanoi, Vietnam. Waste is first collected using handcarts and then transported to collection points at specific intervals. The collection process is therefore guided by inter-arrival time constraints and multiple time windows. Angelelli and Speranza (2002b) apply a modification of Cordeau *et al.*'s (1997) unified tabu search (TS) algorithm to a PVRP-IF with features such as service durations and a maximum tour duration. In Angelelli and Speranza (2002a), this framework is used to analyze the operational cost benefits of different waste collection policies in Val Trompia, Italy and Antwerp, Belgium.

Kim *et al.* (2006) include time windows and a driver break in the collection VRP-IF. Compared to Angelelli and Speranza (2002b), their problem is non-periodic. Yet, they explicitly consider tour compactness and workload balancing. Their solution approach, an extension of Solomon's (1987) insertion algorithm followed by simulated annealing, leads to a significant reduction in the number of tours and substantial financial savings at a major US waste collection company (see Sahoo *et al.,* 2005). Kim *et al.* (2006) are also the first to propose a set of 10 benchmark instances for the VRP-IF, involving up to 2092 stops and 19 intermediate disposal facilities. The multi-objective genetic algorithm of Ombuki-Berman *et al.* (2007), the variable neighborhood tabu search of Benjamin (2011) and the adaptive large neighborhood search (ALNS) of Buhrkal *et al.* (2012) are tested on these instances. Compared to Kim *et al.*'s (2006) results, these algorithms improve average distance by approximately 15% and use fewer vehicles. Buhrkal *et al.*'s (2012) approach also leads to a distance improvement of 30-45% at a Danish waste collection company.

A closely related problem is the multi-depot VRP with inter-depot routes (MDVRPI) proposed by Crevier *et al.* (2007). It is non-periodic, no time windows or driver breaks are considered and, in the general case, depots and intermediate facilities coincide. The authors use the adaptive memory (AM) principle of Rochat and Taillard (1995) and decompose the problem into multi-depot, single-depot and inter-depot subproblems which are solved using Cordeau *et al.*'s (1997) TS. A solution to the MDVRPI is obtained through a set covering problem and optimized by a modified version of the TS.

Crevier *et al.* (2007) observe that when vehicles are stationed at several depots, inter-depot routes occur infrequently possibly because they are rarely economical. Hence, they create two sets of MDVRPI instances with 48 to 288 customers and a fixed homogeneous fleet stationed at one depot, with the rest of the depots acting only as intermediate facilities. These instances are used by Tarantilis *et al.* (2008) and Hemmelmayr *et al.* (2013) who propose, respectively, a hybrid guided local search and a variable neighborhood search (VNS) with a dynamic programming procedure for the insertion of the intermediate facilities in the tours. The former apply the MDVRPI framework to a distribution problem, while the latter to a solid waste collection problem. Both articles report small improvements over the results of Crevier *et al.* (2007) with computation times of only several minutes, even for the largest problems. In addition, Hemmelmayr *et al.* (2013) apply their approach to a PVRP-IF faced by a real waste collection company and obtain a 25% reduction in the routing cost.

Most of the above research concerning intermediate facilities puts unrealistic assumptions on the vehicle fleet. For example, Kim *et al.* (2006) and the related papers on the VRP-IF assume an unlimited homogeneous fleet. The PVRP-IF (Angelelli and Speranza, 2002b) and MDVRPI (Crevier *et al.,* 2007) approaches also assume a homogeneous fleet, albeit limited. However, in industry vehicle fleets are rarely homogeneous. Taillard (1996) was the first to formally define the heterogeneous fixed fleet VRP (HFFVRP). Being a generalization of the vehicle fleet mix problem (VFMP) (e.g. Golden *et al.,* 1984, Salhi *et al.,* 1992), the HFFVRP is NP-hard and more difficult than the classical VRP or the VFMP.

Taillard's (1996) solution approach relies on heuristic column generation with AM. The TS of Taillard (1993) is used to solve an unlimited homogeneous fleet VRP for each vehicle type, with the resulting tours stored in the AM. At each iteration, tours are extracted from the AM favoring those belonging to better solutions, which are then used as a starting point for the TS to obtain another solution. The new tours are added to the AM and dominated tours are removed. Vehicle assignment costs are calculated at each iteration. Finally, the HFFVRP is solved as a set partitioning problem on the tours in the AM. Taillard (1996) adapts the eight largest VFMP instances of Golden *et al.* (1984) to the HFFVRP by removing the vehicle fixed costs and specifying the number of vehicles of each type and their variable costs. More than a dozen authors compete by proposing solutions to these instances and modified versions thereof. Some of the best heuristic approaches are due to Penna *et al.* (2013) and Subramanian *et al.* (2012), the latter also being the fastest. The only fully exact method, which also proves the optimality of seven of the eight best known solutions (BKS), is that of Baldacci and Mingozzi (2009).

Despite the marked research interest it sparked, aside from several case studies and generalizations, the HFFVRP setup of Taillard (1996) is limited in the sense that it does not include some important features such as multiple depots, site dependencies, time windows, driver breaks and maximum tour duration, all of which are present in our waste collection problem. Cordeau *et al.* (2001) were the first to develop a unified TS heuristic for the VRP with time windows (VRPTW) as well as two of its generalizations—the periodic VRPTW (PVRPTW) and multi-depot VRPTW (MDVRPTW). The proposed heuristic is an adaptation of Cordeau *et al.*'s (1997) TS for the non-time window constrained versions of the problems without any major structural and parametric modifications. In Cordeau and Laporte (2001), the same heuristic is also adapted to the site-dependent VRPTW (SDVRPTW).

The presence of time windows leads to some complications in handling maximum tour duration. Cordeau *et al.* (2004) propose a simple refinement of the TS presented in Cordeau *et al.* (2001) for handling several classes of VRPTW with maximum tour duration constraints. The idea is based on forward time slack, a concept introduced by Savelsbergh (1992). Forward time slack keeps track of waiting times and the maximum time a service can be delayed without violating time windows on the tour, thus reducing travel time. This idea is adapted to the specifics of the TS which permits infeasible intermediate solutions. Forward time slack is also exploited by Polacek *et al.* (2004), whose VNS performs slightly better on the same test problems. Contrary to the HFFVRP, however, the last-mentioned problems consider a homogeneous vehicle fleet.

Relatively few authors extend the heterogeneous fixed fleet concept to multiple depots, and many that do it formulation- or methodology-wise, use homogeneous fleet test instances. Dondo and Cerdá (2006) consider a pickup and delivery MDVRPTW and present a reactive mixed integer linear programming (MILP) algorithm that seeks an improvement of the current solution within a feasible region around it. At each iteration, the solution space is restricted by certain pre-assignments of vehicles to depots and to nodes and fixed relative node orderings in some tours, which eliminates many binary variables—an idea similar to large neighborhood search (LNS). Dondo and Cerdá (2007) describe a hybrid cluster-based optimization approach for the same problem. The fleet size is a decision variable but the resulting mix as well as the vehicle-to-depot and vehicle-to-tour assignment becomes an input to the improvement procedure of Dondo and Cerdá (2009). It explores a large neighborhood of the current solution by performing node exchanges among nearby tours and node reordering in each tour. Manageable MILP formulations are developed for both procedures and a spatial decomposition scheme is used to reduce the problem size. Most of the computational tests in the above articles, however, are performed on the single-depot homogeneous fleet instances of Solomon (1987), often failing to reach the BKS even with significant computational effort.

Bettinelli *et al.* (2011) present a branch-and-cut-and-price algorithm for the heterogeneous fleet MDVRPTW, which can be used both as an exact method and as heuristic column generation. On VFMP sets with one and two depots modified from Liu and Shen (1999), the exact approach struggles on problems with more than 25 nodes, while heuristic column generation obtains competitive results. Of the four smallest MDVRPTW instances of Cordeau *et al.* (2001) with fixed homogeneous fleet, only the first two, with 48 and 96 customers, are solved to optimality. Other VRP formulations involving multiple depots and a fixed heterogeneous fleet can be found in Bae *et al.* (2007), Jeon *et al.* (2007) and Wang *et al.* (2008). Goel and Gruhn (2008) describe a non-traditional formulation of a pickup and delivery problem applied to air-cargo road transportation. Requests arrive continuously and are treated at a given time step with a simultaneous decision on which ones to accept and how to route them using a heterogeneous fixed fleet currently servicing other requests. The authors develop a VNS and an LNS and apply them to instances with hundreds of vehicles and requests and a computation time limit of 1 min. Goel (2010) develop an efficient column generation approach for the same problem which outperforms both heuristics with respect to available computation time.

In the last decade or so, the advances in computing power and the development of fast and efficient metaheuristics have made it possible to tackle larger and richer vehicle routing problems. Pisinger and Ropke (2007) use the ALNS of Ropke and Pisinger (2006) to solve several classes of vehicle routing problems, including the VRPTW, the multi-depot VRP (MDVRP), the site-dependent VRP and the open VRP. Each problem variant is appropriately transformed into a rich pickup and delivery problem with time windows, site dependencies, a fixed heterogeneous fleet, and customer visit precedences. At the master level, the search is guided by simulated annealing and at each ALNS iteration an algorithm is chosen to destroy part of the current solution, and another one to repair it. The choice depends on the score the algorithm receives based on past performance, hence the adaptive nature of the search.

Pisinger and Ropke (2007) test their ALNS with a single parameter setting on 486 benchmark instances from various problem classes and improve 183 BKS. In particular, they find 15 new BKS for the 33 MDVRP instances in Cordeau *et al.* (1997) within 4 min of computation time. The latter, however, are based on fixed and unlimited homogeneous fleets and contain no time windows. Baldacci and Mingozzi's (2009) unified exact approach is also tested on nine of the unlimited fleet MDVRP instances with 50 to 160 customers and two to four depots. They obtain ties or slight improvements compared to Pisinger and Ropke (2007) and prove optimality in seven cases. However, their computation times are almost triple. Baldacci and Mingozzi (2009) develop and solve eight more problems with between 150 and 199 customers, three to four depots and an unlimited fleet. For these instances, upper bounds are set by the authors' implementation of the TS algorithm of Cordeau *et al.* (1997). Optimal solutions are obtained for seven of the eight problems, and in only one case is the optimal solution slightly better than the upper bound. Moreover, the implementation of Cordeau *et al.*'s (1997) heuristic is on average 25 times faster than the exact approach.

Ceselli *et al.* (2009) present a rich heterogeneous fixed fleet MDVRP with many non-traditional constraints such as multiple capacities (volume, weight, value), incompatibilities between goods, vehicles, depots and customers, order splitting, multiple tours per day and the possibility of the last daily tours being open tours. The goal is the minimization of a complex hierarchical fee-based cost function. The authors develop a column generation approach where the master problem is a set covering formulation and each column represents a feasible vehicle duty. The pricing step is a resource-constrained elementary shortest path problem, which is solved using the bidirectional bounded dynamic programming algorithm of Righini and Salani (2008) by encoding the constrains in terms of suitable resources. Ceselli *et al.* (2009) remark that, to their knowledge, the only other similar richly constrained problem is the one of Xu *et al.* (2003). The goal of the latter, however, is finding the most efficient fleet composition.

3 Formulation

To our knowledge, the MDVRPI is the only multi-depot framework considering intermediate facilities. Nonetheless, the spirit of our formulation is much closer to that of Kim *et al.* (2006), who consider time windows and a driver break. Similarly, in our case depots and intermediate facilities do not coincide. Therefore, unlike in Crevier *et al.* (2007), trips to the intermediate facilities are bound to occur even when we have vehicles starting from different points. Kim *et al.* (2006) do not provide a mathematical model, but one is presented in Sahoo *et al.* (2005). We should also mention the model of Bard *et al.* (1998a) who consider satellite facilities in a distribution context, but no time windows and a homogeneous fleet. As most authors, we consider the treatment of a single waste material type at a time.

Formally, we define the problem on a directed multigraph $G(O \cup D \cup P, E)$, where O is the set of depots, D is the set of dumps, P is the set of containers, and $E = (i, j : i, j \in O \cup D \cup P, i \neq j)$ is the edge set, with which is associated an asymmetric distance matrix Π, where π_{ij} is the length of edge (i, j). Each vehicle may have a different average speed, which results in a vehicle specific travel time matrix T_k, where τ_{ijk} is the travel time of vehicle k on edge (i, j). Each point has a single time window $[\lambda_i, \mu_i]$, where λ_i and μ_i stand for the earliest and latest possible start-of-service time. Start of service after μ_i is not allowed and if the vehicle arrives before λ_i it has to wait. Since depots are not directly associated with time windows, we consider dummy time windows $[0, \infty]$ for them. Service duration for each point is denoted by ϵ_i, and the pickup quantity volume and weight by ρ_i^v and ρ_i^w, respectively. Service duration at depots and the pickup quantity volume and weight at depots and dumps are fixed at 0.

There is a fixed heterogeneous fleet K, with each vehicle defined by its capacity in terms of maximum volume Ω_k^v and weight Ω_k^w, a fixed cost ϕ_k, a unit-distance running cost β_k and an hourly driver wage rate θ_k. There is a maximum tour duration of H corresponding to the maximum daily work duration, and a break of duration δ must be taken after η hours of continuous work. Site dependencies are described by a binary parameter α_{ijk} whose value is 1 if edge (i, j) is accessible for vehicle k, and 0 otherwise.

In addition, we introduce the following binary decision variables: $x_{ijk} = 1$ if vehicle k traverses edge (i, j), 0 otherwise; $b_{ijk} = 1$ if vehicle k takes a break on edge (i, j), 0 otherwise; $y_k = 1$ if vehicle k is used, 0 otherwise. Three groups of continuous variables, Q_{ik}^v, Q_{ik}^w and S_{ik}, are defined to track the cumulative volume and weight and the start-of-service time at point i for vehicle k. For modeling purposes, we duplicate the set O as O' and O'', the sets of origins and destinations, respectively, and for notation purposes we set $N = O' \cup O'' \cup D \cup P$. Moreover, we assume that each dump in the set D is replicated as many times as the maximum number of dump visits by any vehicle k, which in a realistic case would not be more than a few times. This is necessary since S_{ik} cannot take several distinct values at the same dump i visited by vehicle k. Finally, M is a big number.

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{i \in O'} \sum_{j \in N} x_{ijk} = y_k, \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in D} \sum_{j \in O''} x_{ijk} = y_k, \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{ijk} = 0, \qquad \forall k \in K, j \in O' \tag{5}$$

$$\sum_{j \in N} x_{ijk} = 0, \qquad \forall k \in K, i \in O'' \tag{6}$$

$$\sum_{i \in N \setminus O''} x_{ijk} = \sum_{i \in N \setminus O'} x_{jik}, \qquad \forall k \in K, j \in D \cup P \tag{7}$$

$$x_{ijk} \leqslant \alpha_{ijk}, \qquad \forall k \in K, i \in N \setminus O'', j \in N \setminus O' \tag{8}$$

$$Q^v_{ik} \leqslant \Omega^v_k, \qquad \forall k \in K, i \in P \tag{9}$$

$$Q^w_{ik} \leqslant \Omega^w_k, \qquad \forall k \in K, i \in P \tag{10}$$

$$Q^v_{ik} = 0, \qquad \forall k \in K, i \in N \setminus P \tag{11}$$

$$Q^w_{ik} = 0, \qquad \forall k \in K, i \in N \setminus P \tag{12}$$

$$Q^v_{ik} + \rho^v_j \leqslant Q^v_{jk} + \left(1 - x_{ijk}\right) M, \qquad \forall k \in K, i \in N \setminus O'', j \in P \tag{13}$$

$$Q^w_{ik} + \rho^w_j \leqslant Q^w_{jk} + \left(1 - x_{ijk}\right) M, \qquad \forall k \in K, i \in N \setminus O'', j \in P \tag{14}$$

$$S_{ik} + \epsilon_i + \delta b_{ijk} + \tau_{ijk} \leqslant S_{jk} + \left(1 - x_{ijk}\right) M, \qquad \forall k \in K, i \in N \setminus O'', j \in N \setminus O' \tag{15}$$

$$\left(S_{ik} - \sum_{m \in O'} S_{mk} \right) + \epsilon_i - \eta \leqslant \left(1 - b_{ijk}\right) M, \qquad \forall k \in K, i \in N \setminus O'', j \in N \setminus O' \tag{16}$$

$$\eta - \left(S_{jk} - \sum_{m \in O'} S_{mk} \right) \leqslant \left(1 - b_{ijk}\right) M, \qquad \forall k \in K, i \in N \setminus O'', j \in N \setminus O' \tag{17}$$

$$b_{ijk} \leqslant x_{ijk}, \qquad \forall k \in K, i, j \in N \tag{18}$$

$$\left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) - \eta \leqslant \left(\sum_{\substack{i \in N \setminus O'' \\ j \in N \setminus O'}} b_{ijk} \right) M, \qquad \forall k \in K \tag{19}$$

$$\lambda_i \sum_{j \in N \setminus O'} x_{ijk} \leqslant S_{ik} \leqslant \mu_i \sum_{j \in N \setminus O'} x_{ijk}, \qquad \forall k \in K, i \in N \setminus O'' \tag{20}$$

$$\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \leqslant \mathrm{H}, \qquad \forall k \in K \tag{21}$$

$$x_{ijk}, y_k, b_{ijk} \in \{0, 1\}, \qquad \forall k \in K, i, j \in N \tag{22}$$

$$Q^v_{ik}, Q^w_{ik}, S_{ik} \geqslant 0, \qquad \forall k \in K, i \in N \tag{23}$$

The objective function (1) minimizes the spatial and temporal costs of the routing schedule. Constraints (2) impose that each collection point should be served exactly once by one vehicle. Equations (3) ensure that each vehicle tour starts at an origin, while equations (4) stipulate that there is a trip from a dump to a destination. Constraints (5) and (6) forbid returning to an origin or leaving a destination. Flow conservation is ensured by constraints (7) and site dependencies are enforced by constraints (8). Inequalities (9) and (10) capacitate, respectively, the cumulative volume and weight on the vehicle at each point, while equations (11) and (12) reset them to zero at the dumps, origins and destinations. Keeping track of the cumulative volume and weight on the vehicle is achieved by constraints (13) and (14). Constraints (15) calculate the start-of-service time at every point, including service duration and a possible break duration. Inequalities (16), (17) and (18) calculate the edge on which a break should be taken while inequalities (19) ensure that the break is actually taken if the vehicle tour is longer than the maximum continuous work limit η. Time window restrictions are captured by constraints (20). Constraints (21) provide that all points must be visited within the maximum tour duration. Finally, (22) and (23) establish the variable domains.

The above formulation introduces several extensions to the model of Sahoo *et al.* (2005), including multiple origins and destinations, multiple capacities, site dependencies, a maximum tour duration, a richer objective function capturing the costs faced by a realistic firm, and the elimination of the constraints calculating the necessary number of disposal trips for each vehicle. The break is modeled in a way similar to Buhrkal *et al.* (2012) but without imposing a hard time window. In fact, constraints (16) to (19) can easily be modified to include multiple breaks after each η duration depending on regulation. The resulting formulation is reminiscent of a location-routing problem since we are looking both for the vehicles' origins and destinations, and their optimal routing. However, by a simple modification of constraints (3) we can fix the vehicles' origins. Regarding the destinations, we prefer to keep the current notation due to the fact that our case study includes a wide service area with a mix of urban and rural regions and it is not always optimal for a vehicle to return to the depot it started from. However, we can provide an incentive to the latter by modifying the model as follows:

$$\text{Min} \quad f = \text{Objective (1)} + \Psi \sum_{k\in K} \sum_{i\in O'} \sum_{j\in O''} \left(\beta_k \pi_{ji} + \theta_k \tau_{jik}\right) z_{ijk} \tag{24}$$

s.t. Constraints (2) to (23)

$$\sum_{m\in P} x_{imk} + \sum_{m\in D} x_{mjk} - 1 \leqslant z_{ijk}, \qquad \forall k \in K, i \in O', j \in O'' \tag{25}$$

$$z_{ijk} = \{0, 1\}, \qquad \forall k \in K, i \in O', j \in O'' \tag{26}$$

where constraints (25) require that the variable z_{ijk} should be equal to 1 if i and j are, respectively, the origin and destination of vehicle k. Obviously, if i and j do not represent the same point, there will be a positive distance and travel time between them. This is captured by the objective (24) which, in addition to (1), minimizes the cost of returning from j to i. The coefficient Ψ determines the importance of the new objective function term.

4 Solution Approach

The vehicle routing problem is well known to be NP-hard (see e.g. Garey and Johnson, 1979). Being a generalization thereof, our waste collection problem is even harder to solve. Moreover, realistic instances involving 50 or more containers and several depots, dumps and vehicles will translate into thousands of binary variables and tens of thousands of constraints. Therefore, for such cases we propose a heuristic approach based on local search. To evaluate the quality of the heuristic, we benchmark the solutions it provides to those obtained by the model presented in Section 3 on small problem instances. In what follows here, we develop several classes of valid inequalities and elimination rules which help improve the root relaxation of the mathematical formulation and fix many binary variables, thus speeding up the branch and bound process. Afterwards, we explain the main ideas behind the local search algorithm.

4.1 Valid Inequalities and Elimination Rules

The first set of rules is used to set to zero binary variables linked to impossible traversals. Constraints (27) eliminate the possibility of loops. In a similar fashion, constraints (28), (29) and (30) forbid traveling from an origin to a dump or destination, from a container to a destination, and from a dump to another dump, respectively.

$$\begin{aligned}
\text{s.t.}\quad & x_{iik} = 0, && \forall k \in K, i \in N && (27)\\
& x_{ijk} = 0, && \forall k \in K, i \in O', j \in D \cup O'' && (28)\\
& x_{ijk} = 0, && \forall k \in K, i \in P, j \in O'' && (29)\\
& x_{ijk} = 0, && \forall k \in K, i \in D, j \in D\colon i \neq j && (30)
\end{aligned}$$

The presence of time windows allows us to fix time-window infeasible traversals. Constraints (31) express the fact that if by servicing point i as early as possible vehicle k cannot service point j within its time window, then points i and j cannot be serviced by the same vehicle k, i.e. edge (i, j) is not traversed by vehicle k. These first two sets of rules can also be used to eliminate all the big M constraints (13, 14, 15, 16, 17, 19) for such variables as they become trivial.

$$\text{s.t.}\quad x_{ijk} = 0, \qquad \forall k \in K, i \in P \cup D, j \in P \cup D\colon \lambda_i + \epsilon_i + \tau_{ijk} > \mu_j \tag{31}$$

The third set of rules serves for restricting the start-of-service time search space. This set of rules has a significant impact on the lower bound of the objective function (24), given the way tour duration is minimized. Rule (32) imposes a minimum difference between the start-of-service time at the origin and destination for each used vehicle. The right-hand side of the inequality represents the minimum-duration tour composed of one origin, one container, one dump and one destination. Then rules (33) and (34) calculate the latest possible start and earliest possible finish of each tour.

$$\text{s.t.}\quad \sum_{j\in O''} S_{jk} - \sum_{i\in O'} S_{ik} \geqslant \min_{\substack{m_1\in O'\\ m_2\in P\\ m_3\in D\\ m_4\in O''}} (\tau_{m_1m_2k} + \epsilon_{m_2} + \tau_{m_2m_3k} + \epsilon_{m_3} + \tau_{m_3m_4k})\, y_k, \qquad \forall k \in K \tag{32}$$

$$S_{ik} \leqslant \max_{m\in P} (\mu_m - \tau_{imk})\, y_k, \qquad \forall k \in K, i \in O' \tag{33}$$

$$S_{jk} \geqslant \min_{m\in D} \left(\lambda_m + \epsilon_m + \tau_{mjk}\right) \sum_{m\in D} x_{mjk}, \qquad \forall k \in K, j \in O'' \tag{34}$$

If the problem involves subsets of identical vehicles, the presence of symmetry can substantially reduce the effectiveness of the model. For example, there are two ways to assign two tours to two identical vehicles, both producing the same value of the objective function. If we have two tours and three identical vehicles, the number of permutations is six. To remove the symmetry, we can use so-called symmetry breaking constraints. Let $K' \subset K$ represent a subset of identical vehicles and let $k'_g \in K'$, where $g \in 1, \ldots, |K'|$ introduces a simple ordering of the elements of K'. Then for each subset K' we apply constraints (35) or (36). These constraints specify that the first vehicle in K' executes the tour with the highest waste volume (weight), the second vehicle executes the tour with the second highest waste volume (weight), etc.

$$\text{s.t.}\quad \sum_{i\in P} \sum_{j\in P\cup D} \rho_i^v x_{ijk'_g} \geqslant \sum_{i\in P} \sum_{j\in P\cup D} \rho_i^v x_{ijk'_{g+1}}, \qquad \forall g \in 1, \ldots, (|K'| - 1) \tag{35}$$

$$\sum_{i\in P} \sum_{j\in P\cup D} \rho_i^w x_{ijk'_g} \geqslant \sum_{i\in P} \sum_{j\in P\cup D} \rho_i^w x_{ijk'_{g+1}}, \qquad \forall g \in 1, \ldots, (|K'| - 1) \tag{36}$$

The last two rules are concerned with the dump visits. With (37) we impose that a dump may be visited at most once by a vehicle. With (38), on the other hand, we set for every vehicle the maximum number of trips from dumps to containers, which is one less the total number of dumps. A dump visit is thus reserved for the final trip in each tour. We stress again here that each physical dump is replicated in the model as many times as the maximum number of dump visits by any vehicle k.

$$\text{s.t.} \quad \sum_{i \in P} x_{ijk} \leqslant 1, \qquad \forall k \in K, j \in D \tag{37}$$

$$\sum_{i \in D} \sum_{j \in P} x_{ijk} \leqslant |D| - 1, \qquad \forall k \in K \tag{38}$$

4.2 Local Search

The heuristic we develop for the problem is based on the idea of local search, which is a systematic exploration of the immediate neighbors of an incumbent solution through a set of predefined neighborhood operators. In our approach, all changes applied to a tour preserve feasibility. In part, this is motivated by the fact that a tour can contain several dump visits. Once we start exploring infeasible neighbors, we may successively violate the vehicle capacities at numerous points linked to disposals at different dumps. A penalty term proportional to the number of visited infeasible solutions will start driving the search towards feasibility again. However, even only considering capacity feasibility, the tour will have to recover it at numerous points, which could be very cumbersome. Moreover, such a technique will involve much additional parameter tuning (see e.g. Cordeau *et al.,* 2001).

A solution to our problem is considered feasible if it satisfies three criteria. First, start-of-service times should respect time windows, where applicable. Secondly, the solution has to have a feasible duration, i.e. its duration should be shorter than or equal to the maximum tour duration defined by H in Section 3. These two criteria may be thought of as expressing temporal feasibility. Thirdly, the volume and weight capacity of the vehicle may not be violated at any point in the tour. This can be ensured by inserting appropriate visits to the available dumps.

Every insertion or removal of a point from a tour, and every application of a neighborhood operator requires the recalculation of start-of-service and waiting times for all or part of the points in the tour. Consider a tour serviced by vehicle $k \in K$, for brevity tour k, represented as an ordered sequence of points $1, \ldots, n$ indexed by i. The calculation begins by setting the start-of-service time at the origin, S_{1k}, as early as possible. For each subsequent point i, S_{ik} is tentatively calculated as the sum of the start-of-service time at point $i-1$, the service duration at point $i-1$, and the travel time from $i-1$ to i, i.e. $S_{ik} = S_{(i-1)k} + \epsilon_{i-1} + \tau_{(i-1)ik}$. If the maximum continuous working time limit η expires between the end-of-service time (start-of-service time plus service duration) at points $i-1$ and i, i.e. if $S_{(i-1)k} + \epsilon_{i-1} \leqslant S_{1k} + \eta$ and $S_{ik} + \epsilon_i > S_{1k} + \eta$, we need to insert the required break before servicing point i, which is achieved by incrementing S_{ik} by the break duration δ. Finally, if S_{ik} violates the lower time window bound λ_i, in other

Figure 2: Temporal feasibility algorithm

```
Data: tour $k$ as a sequence of points $1, \dots, n$ after the insertion or removal of a point or the
      application of a neighborhood operator
Result: start-of-service times, waiting times and temporal feasibility of tour $k$
set $S_{1k}$ to earliest possible;
for $i = 2 \dots n$ in tour $k$ do
    $S_{ik} = S_{(i-1)k} + \epsilon_{i-1} + \tau_{(i-1)ik}$;
    if $S_{(i-1)k} + \epsilon_{i-1} \leqslant S_{1k} + \eta$ and $S_{ik} + \epsilon_i > S_{1k} + \eta$ then
        $S_{ik} = S_{ik} + \delta$;
    end
    if $S_{ik} < \lambda_i$ then
        $w_{ik} = \lambda_i - S_{ik}$;
        $S_{ik} = \lambda_i$;
    else
        $w_{ik} = 0$;
    end
end
if $S_{ik} \leqslant \mu_i, \forall i$ then
    for $i = n \dots 2$ in tour $k$ do
        $S'_{(i-1)k} = S_{(i-1)k}$;
        $S_{(i-1)k} = \min(S_{(i-1)k} + w_{ik}, \mu_{i-1})$;
        $w_{(i-1)k} = w_{(i-1)k} + \left(S_{(i-1)k} - S'_{(i-1)k}\right)$;
        $w_{ik} = w_{ik} - \left(S_{(i-1)k} - S'_{(i-1)k}\right)$;
    end
    $w_{1k} = 0$;
    if $S_{nk} - S_{1k} \leqslant \mathrm{H}$ then
        tour $k$ is temporally feasible;
    else
        discard tour $k$ as duration infeasible;
    end
else
    discard tour $k$ as time-window infeasible;
end
```

words if $S_{ik} < \lambda_i$, we introduce waiting time w_{ik} at point i equal to the difference $\lambda_i - S_{ik}$, and update S_{ik} to λ_i. Once all S_{ik} have been determined, we check if upper time window bounds

μ_i are respected for all i. If this is the case, we apply forward time slack reduction on the tour, otherwise we discard the tour as time-window infeasible.

Forward time slack, as described by Savelsbergh (1992), keeps track of the maximum amount each start-of-service time can be delayed without violating time windows on the tour. Under the assumption that we initialize S_{1k} as early as possible and introduce waiting times due to time windows, pushing S_{1k} forward may, in extreme cases, eliminate all waiting. Realistically, we will obtain a tour whose duration is at most that of the original tour. The idea is very similar to the concept of slack in the Project Evaluation and Review Technique (PERT), which indicates how much a task in the project can be delayed without causing any delay in subsequent tasks.

To apply forward time slack reduction, points on the tour are examined sequentially in reverse order. If there is waiting at point i, there could be a non-zero slack at point $i-1$, because pushing $S_{(i-1)k}$ forward may eliminate or reduce waiting at i. We can push $S_{(i-1)k}$ forward by the amount of waiting at i, or until we reach the upper time window bound at $i-1$. The last operation is expressed as $S_{(i-1)k} = \min(S_{(i-1)k} + w_{ik}, \mu_{i-1})$, and it entails an update of $w_{(i-1)k}$ and w_{ik} to factor in the potential increase of waiting at $i-1$ and decrease of waiting at i. Let $S'_{(i-1)k}$ denote the original start-of-service time at point $i-1$, i.e. before slack reduction. Then, waiting at $i-1$ will be increased by the difference between $S_{(i-1)k}$ and $S'_{(i-1)k}$, and waiting at i will be reduced by the same difference. Finally, we need to artificially put $w_{1k} = 0$. Forward time slack reduction preserves time-window feasibility. Therefore, after the procedure it only remains to check if the tour's duration is feasible. If it is the case, we accept the tour as temporally feasible, otherwise we discard it as duration infeasible. Figure 2 is a schematic depiction of the procedure for calculating start-of-service and waiting times, and verifying temporal feasibility.

The last feasibility criterion, concerning capacity, is much more straightforward. At each point of the tour, we calculate the cumulative volume and weight loads, Q^v_{ik} and Q^w_{ik}, on the vehicle, resetting both to zero if the point is a dump. If, for any point i, $Q^v_{ik} > \Omega^v_k$ or $Q^w_{ik} > \Omega^w_k$ or a dump is not visited immediately before the destination, we discard the tour as capacity infeasible.

4.2.1 Construction

Tour construction is performed sequentially with the goal of minimizing the objective function (24). Initially, all containers belong to the pool of unassigned containers P, and all vehicles to the pool of unassigned vehicles K. A seed tour is created by assigning the cheapest sequence of origin, container, dump and destination to the cheapest available vehicle. We assume here that, even in the presence of time windows, each container can participate in at least one feasible seed tour sequence and that the smallest vehicle's capacity is sufficient to service the largest container when the latter is full. Such an assumption should hold in any realistic scenario. It is not a particularity of the heuristic approach but is also assumed by the mathematical model. All assigned vehicles and containers are removed from their respective unassigned pools.

Once a seed tour k has been created, it is expanded using a simple feasibility preserving greedy insertion heuristic. At each iteration, we insert container $i \in P$ at the position j in the tour that would yield the smallest increase in the tour's objective value. The point at position j, as well as all subsequent points, are shifted to the right. As already mentioned, we require that such insertions should produce a feasible tour. If no more feasible container insertions are possible and if infeasibility would result from capacity violation, we insert a dump using the same logic, otherwise we terminate the tour. In addition, we require that the dump cannot be inserted as an immediate predecessor or successor of another dump on the tour or just after the origin depot. Unlike in the mathematical model, here we do not need to copy a dump object as many times as it can be visited during a tour.

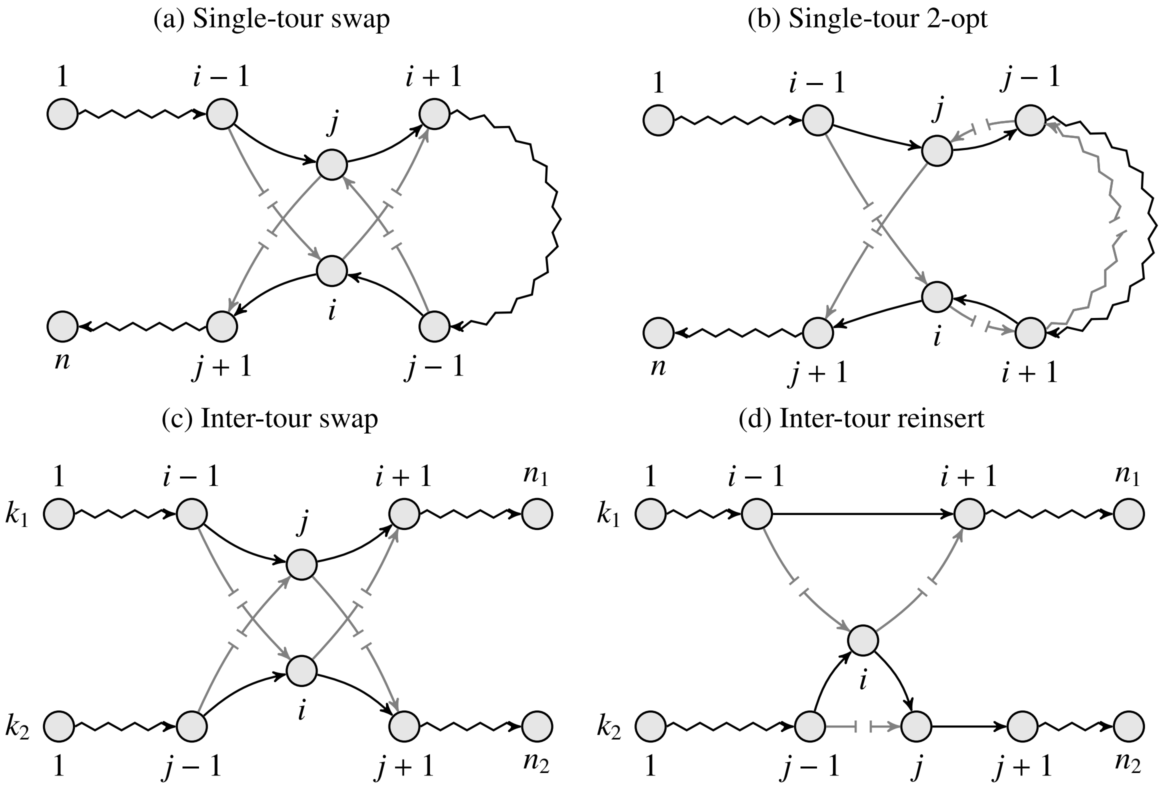
Finally, to avoid a meaningless increase in the objective function, we require that after a dump insertion there should be a predefined number of feasible container insertions. If this condition does not hold, the last inserted dump is removed and the tour is terminated. Tour construction stops when either the pool of unassigned containers or the pool of unassigned vehicles is empty.

4.2.2 Improvement

Tour improvement makes use of four neighborhood operators—single-tour swap and 2-opt and inter-tour swap and reinsert. Single-tour operators are applied to points or collections of points belonging to the same tour, while inter-tour operators are applied to points belonging to two different tours. All operators are feasibility preserving. If the application of an operator would result in an infeasible tour, it is not carried out. Figure 3 depicts possible improvements resulting from the application of each of the operators. The interrupted gray edges form parts of the tours before the application of the operators. The resulting improved tours are given in solid black edges.

Consider a tour as an ordered sequence of points represented by $1, \dots, n$ and two points, i and j, belonging to the tour, where $i + 1 < j - 1$. A single-tour swap disconnects $i - 1$ from i, i from $i + 1$, $j - 1$ from j, and j from $j + 1$, and reconnects $i - 1$ to j, j to $i + 1$, $j - 1$ to i, and i to $j + 1$. A single-tour 2-opt operator, on the other hand, disconnects $i - 1$ from i, and j from $j + 1$, and reconnects $i - 1$ to j, and i to $j + 1$, thus reversing the orientation of the section $i, \dots, j$, inclusive of i and j.

Figure 3: Neighborhood operators


(a) Single-tour swap

(b) Single-tour 2-opt

(c) Inter-tour swap

(d) Inter-tour reinsert

To illustrate the inter-tour operators, consider two tours k_1 and k_2, a point i in k_1 and a point j in k_2. The inter-tour swap operator applies the same logic as the single-tour one, except that here points i and j belong to different tours. With the inter-tour reinsert operator, point i is removed from tour k_1, thus connecting $i-1$ directly to $i+1$. It is then inserted in k_2 at the position of point j, which is achieved by disconnecting $j-1$ from j, and reconnecting it to i, and then connecting i to j.

Tour improvement itself is an alternation between single-tour and inter-tour improvement. Each single-tour operator is applied for *maxOpIter* iterations and *maxOpNonImpIter* non-improving iterations from the last visited cheapest incumbent. Moreover, each operator is applied to the i and j that would result in the best feasible neighbor of the incumbent. Therefore, we change the operator in one of three cases—(a) we reached the maximum number of iterations, (b) the maximum number of non-improving iterations or (c) the operator cannot produce feasible neighbors of the incumbent. To prevent cycling and encourage diversification towards less explored areas of the search space, a solution with the same objective value is not admitted more than once for a given number of iterations. The couple of single-tour operators is applied successively for *maxIter* iterations and *maxNonImpIter* non-improving iterations. Tour improvement can make a tour a candidate for more container insertions if unassigned containers remain in the pool. New insertions are attempted before switching back from single-tour 2-opt to single-tour swap.

When all tours have been constructed and improved, inter-tour swap and reinsert are applied exhaustively on all tour combinations using the same logic with the same iteration limiters, where this time an operator is applied to the two tours and two points that would produce the best feasible neighbor of the tour schedule. After the application of an inter-tour operator, both affected tours are individually improved. In some cases, one of the tours may need one fewer dump visits, in which case instead of being simply improved it is dissolved and recreated from scratch.

5 Case Study

Our case study considers a French waste collection company, which operates more than 2800 containers located at 820 collection points spread out over a large rural and peri-urban area of approximately 2000 km^2. Each container holds one of three types of recyclable material—paper, glass or plastic/aluminum. Most containers are equipped with level sensors that transmit the volume of waste in the container via the GSM network. The weight of the respective waste type is calculated through a conversion algorithm. The distance matrix between all depots, dumps and collection points has been calculated with the OsmSharp[1] routing library on OpenStreetMap[2] data.

To assess the quality of the local search heuristic, we compare its results to the optimal ones produced by the mathematical model on small instances. The instances are constructed randomly but based on real underlying data. We create five instances, each made up of 10 containers of the same waste material type, one depot and one dump. The original instances include no time windows. Two additional copies of each instance are therefore created, one with wide time windows and one with narrow time windows. Wide time windows are created by randomly assigning to each container a time window of [8:00, 12:00] or [12:00, 18:00] with equal probability. Narrow time windows are created in the same fashion and are represented by the intervals [8:00, 11:00], [12:00, 15:00] and [15:00, 17:00]. In addition, we restrict the tour to a maximum duration of eight hours with a one-hour break after four hours of continuous work. It takes five min to service a container and 10 min to empty the vehicle at a dump.

Regarding the vehicles, for each instance we create two homogeneous trucks with volume and

[1] See http://www.osmsharp.com, accessed on April 20, 2014.
[2] See http://download.geofabrik.de, accessed on April 20, 2014.

Table 1: Comparison between the local search heuristic and the Gurobi MIP solver on random instances
maxOpIter = 100, *maxOpNonImpIter* = 13, *maxIter* = 100, *maxNonImpIter* = 1

		Heuristic		Solver					
Instance	Tours	Objective	Runtime (sec.)	Objective	Lower bound	MIP gap (%)	Relaxation	Runtime (sec.)	Opt gap (%)
i1	1	214.849	0.170	214.849	214.837	0.006	11.250	375.562	0.000
i1_tw	1	284.016	0.070	252.825	252.825	0.000	95.627	4.038	12.337
i1_ntw	2	428.539	1.093	394.817	394.817	0.000	179.195	0.922	8.541
i2	1	249.317	0.042	249.317	249.317	0.000	58.791	400.032	0.000
i2_tw	1	257.583	0.050	257.582	257.582	0.000	119.750	2.306	0.000
i2_ntw	2	460.635	0.756	439.769	439.769	0.000	235.777	2.420	4.745
i3	1	240.133	0.051	240.133	76.004	68.349	14.926	1000.000	0.000
i3_tw	1	245.457	0.070	245.457	245.457	0.000	45.625	2.894	0.000
i3_ntw	2	444.589	0.641	444.589	444.589	0.000	76.527	2.446	0.000
i4	1	138.643	0.077	138.643	138.643	0.000	4.077	521.509	0.000
i4_tw	1	140.204	0.030	140.204	140.204	0.000	4.077	7.660	0.000
i4_ntw	1	179.537	0.043	179.537	179.537	0.000	23.208	2.849	0.000
i5	1	220.770	0.070	220.770	129.834	41.190	37.889	1000.000	0.000
i5_tw	1	233.211	0.050	233.211	233.211	0.000	83.942	3.501	0.000
i5_ntw	2	405.622	0.848	405.622	405.622	0.000	106.033	3.051	0.000

Table 2: Comparison between the local search heuristic and the Gurobi MIP solver on selected random instances $maxOpIter$ = 350, $maxOpNonImpIter$ = 37, $maxIter$ = 100, $maxNonImpIter$ = 1

		Heuristic		Solver					
Instance	Tours	Objective	Runtime (sec.)	Objective	Lower bound	MIP gap (%)	Relaxation	Runtime (sec.)	Opt gap (%)
i1_tw	1	252.825	0.410	252.825	252.825	0.000	95.627	3.487	0.000
i1_ntw	2	394.817	3.399	394.817	394.817	0.000	179.195	0.916	0.000
i2_ntw	2	439.769	3.080	439.769	439.769	0.000	235.777	2.309	0.000

weight capacity sufficient to ensure that the containers in all instances can be collected with only one dump visit before the end of the tour in the absence of time windows. We assume that a vehicle k's fixed cost ϕ_k is five times its unit distance running cost β_k, and its driver wage rate is 10 times β_k.

All tests were carried out on a 2.60 GHz Intel Core i7 PC with 8 GB of memory running a 64-bit Windows 7. The local search heuristic was coded in Java and the mathematical model was solved using the Gurobi 5.6.2 MIP solver warm-started, for each instance, with the solution obtained by the heuristic. Table 1 presents the results of the comparison for the 15 instances, where "tw" in the instance name stands for time windows, and "ntw" for narrow time windows. The second column displays the number of tours constructed. Then, for the heuristic, the table reports the objective value and runtime in seconds achieved for the parameters in the table subtitle. For the solver, it reports, in addition, the lower bound, MIP gap and root relaxation. The solver time limit was set to 1000 sec. The last column in the table is the heuristic's optimality gap relative to the objective found by the solver within the time limit. Since there are no random components in the heuristic, all results are presented for one single run.

The table reveals that our local search heuristic performs reasonably well with an average optimality gap of 1.7% and a runtime of a fraction of a second, except for the instances with narrow time windows, where it still remains in the order of 1 sec. There is a non-zero optimality gap for only three instances, all of which contain time windows. The presence of time windows also appears to increase the heuristic's runtime, probably because of slower convergence due to many more infeasible neighbors in the improvement stage. Looking at the results for the solver, we can observe the opposite phenomenon with runtime negatively correlated to the presence of time windows and their tightness. This is explained by the fact that time windows allow for many binary variables to be fixed to zero, thus greatly reducing the search space for branch and bound. For two instances without time windows, the solver is unable to prove optimality within 1000 sec. This is due to the poor values of the root relaxation, undoubtedly the result of the big M constraints in the mathematical formulation. Finally, Table 2 shows that at the expense of more iterations the heuristic can reach the optimal solutions to the instances that have a non-zero optimality gap in Table 1.

6 Conclusion

This article proposed a mathematical model and a local search heuristic for a complex solid waste collection problem, which can be viewed as a generalization of the vehicle routing problem with intermediate facilities (Kim *et al.,* 2006). We add several new features such as a realistic cost-based objective function, multiple depots, a fixed heterogeneous fleet, site dependencies, a start-of-tour dependent driver break, and a relocation cost, which incentivizes, rather than enforcing, the vehicle to return to the depot it started from. The performance of the local search heuristic was evaluated by comparison to the optimal solution on small instances of 10 containers and varying time window tightness. Preliminary results show that the heuristic performs favorably with an average optimality gap of less that 2% and an runtime of less than 1 sec.

Future work will focus on several topics, the first of which is an attempt at reformulation of the mathematical model, especially the big M constraints. Currently, the root relaxation provides a very weak bound which, in the absence of time windows, leads to very slow convergence of the branch and bound process. A promising direction proposed in the literature is to express some of the big M constraints, for example the capacity constraints, as flows. Here the approach has to be adapted to the particular fact that capacity in our problem can be renewed by a visit to a dump. At any rate, the ability to solve larger and more complex instances to optimality is important in terms of providing more meaningful benchmarks for the performance of the local search heuristic.

Secondly, there are several important improvements to be added to the current local search heuristic. When a seed tour is created, its origin and destination are no longer changed during the construction and improvement stage. Once inserted, neither are the dumps that are visited during the tour, unless an inter-tour swap exchanges two different dumps from two different tours. Since a problem may contain several available depots and dumps, the construction and improvement stages of the algorithm need to examine whether the growth and improvement of the tour would benefit from depot or dump reassignments. Regarding the heterogeneity of the vehicles, the design of new and efficient inter-tour operators is an important next step in ensuring a good vehicle-to-tour assignment. Last but not least, a thorough sensitivity analysis of the heuristic parameters needs to be carried out.

Finally, however much the mathematical formulation can be improved, it would not be suitable for solving real world instances involving tens or hundreds of containers as well as several depots and dumps. In these cases, we can benchmark the heuristic solutions against the current state of practice at the French waste collection company where, since recently, all executed tours are being logged together with the amount of material emptied from each container. Such comparisons will allows us to asses the heuristic's added value to the company.

Acknowledgement

This work is partially funded by the Commission for Technology and Innovation (CTI) under grant number CTI 15781.1 PFES-ES. The authors would also like to thank EcoWaste SA (http://www.ecowaste.ch/) for their expert contribution and the problem data.

7 References

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

Angelelli, E. and M. G. Speranza (2002b) The periodic vehicle routing problem with intermediate facilities, *European Journal of Operational Research*, **137** (2) 233–247.

Bae, S.-T., H. S. Hwang, G.-S. Cho and M.-J. Goan (2007) Integrated GA-VRP solver for multi-depot system, *Computers & Industrial Engineering*, **53** (2) 233–240.

Baldacci, R. and A. Mingozzi (2009) A unified exact method for solving different classes of vehicle routing problems, *Mathematical Programming*, **120** (2) 347–380.

Bard, J. F., L. Huang, M. Dror and P. Jaillet (1998a) A branch and cut algorithm for the VRP with satellite facilities, *IIE Transactions*, **30** (9) 821–834.

Bard, J. F., L. Huang, P. Jaillet and M. Dror (1998b) A decomposition approach to the inventory routing problem with satellite facilities, *Transportation Science*, **32** (2) 189–203.

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

Benjamin, A. M. (2011) Metaheuristics for the waste collection vehicle routing problem with time windows, Ph.D. Thesis, Department of Mathematical Sciences, Brunel University.

Bettinelli, A., A. Ceselli and G. Righini (2011) A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows, *Transportation Research Part C: Emerging Technologies*, **19** (5) 723–740.

Bhat, V. N. (1996) A model for the optimal allocation of trucks for the solid waste management, *Waste Management & Research*, **14** (1) 87–96.

Buhrkal, K., A. Larsen and S. Ropke (2012) The waste collection vehicle routing problem with time windows in a city logistics context, *Procedia-Social and Behavioral Sciences*, **39**, 241–254.

Ceselli, A., G. Righini and M. Salani (2009) A column generation algorithm for a rich vehicle-routing problem, *Transportation Science*, **43** (1) 56–69.

Clarke, G. and J. W. Wright (1964) Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, **12**, 568–581.

Cordeau, J.-F., M. Gendreau and G. Laporte (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems, *Networks*, **30** (2) 105–119.

Cordeau, J.-F. and G. Laporte (2001) A tabu search algorithm for the site dependent vehicle routing problem with time windows, *INFOR*, **39** (3) 292–298.

Cordeau, J.-F., G. Laporte and A. Mercier (2001) A unified tabu search heuristic for vehicle routing problems with time windows, *The Journal of the Operational Research Society*, **52** (8) 928–936.

Cordeau, J.-F., G. Laporte and A. Mercier (2004) Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows, *The Journal of the Operational Research Society*, **55**, 542–546.

Crevier, B., J.-F. Cordeau and G. Laporte (2007) The multi-depot vehicle routing problem with inter-depot routes, *European Journal of Operational Research*, **176** (2) 756–773.

Dondo, R. and J. Cerdá (2006) A reactive MILP approach to the multidepot heterogeneous fleet vehicle routing problem with time windows, *International Transactions in Operational Research*, **13** (5) 441–459.

Dondo, R. and J. Cerdá (2007) A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows, *European Journal of Operational Research*, **176** (3) 1478–1507.

Dondo, R. G. and J. Cerdá (2009) A hybrid local improvement algorithm for large-scale multi-depot vehicle routing problems with time windows, *Computers & Chemical Engineering*, **33** (2) 513–530.

Garey, M. R. and D. S. Johnson (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA.

Goel, A. (2010) A column generation heuristic for the general vehicle routing problem, in C. Blum and R. Battiti (eds.) *Learning and Intelligent Optimization*, vol. 6073 of *Lecture Notes in Computer Science*, 1–9, Springer Berlin Heidelberg.

Goel, A. and V. Gruhn (2008) A general vehicle routing problem, *European Journal of Operational Research*, **191** (3) 650–660.

Golden, B., A. Assad, L. Levy and F. Gheysens (1984) The fleet size and mix vehicle routing problem, *Computers & Operations Research*, **11** (1) 49–66.

Hemmelmayr, V., K. F. Doerner, R. F. Hartl and S. Rath (2013) A heuristic solution method for node routing based solid waste collection problems, *Journal of Heuristics*, **19** (2) 129–156.

Jeon, G., H. R. Leep and J. Y. Shim (2007) A vehicle routing problem solved by using a hybrid genetic algorithm, *Computers & Industrial Engineering*, **53** (4) 680–692.

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

Liu, F.-H. and S.-Y. Shen (1999) The fleet size and mix vehicle routing problem with time windows, *The Journal of the Operational Research Society*, **50** (7) 721—732.

Ombuki-Berman, B. M., A. Runka and F. T. Hanshar (2007) Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms, paper presented at the *Proceedings of the Third IASTED International Conference on Computational Intelligence*, CI ’07, 91–97, Anaheim, CA, USA.

Penna, P., A. Subramanian and L. Ochi (2013) An iterated local search heuristic for the heterogeneous fleet vehicle routing problem, *Journal of Heuristics*, **19** (2) 201–232.

Pisinger, D. and S. Ropke (2007) A general heuristic for vehicle routing problems, *Computers & Operations Research*, **34** (8) 2403–2435.

Polacek, M., R. F. Hartl, K. Doerner and M. Reimann (2004) A variable neighborhood search for the multi depot vehicle routing problem with time windows, *Journal of Heuristics*, **10** (6) 613–627.

Righini, G. and M. Salani (2008) New dynamic programming algorithms for the resource constrained elementary shortest path problem, *Networks*, **51** (3) 155–170.

Rochat, Y. and E. Taillard (1995) Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics*, **1**, 147–167.

Ropke, S. and D. Pisinger (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, **40** (4) 455–472.

Sahoo, S., S. Kim, B.-I. Kim, B. Kraas and A. Popov (2005) Routing optimization for waste management, *Interfaces*, **35** (1) 24–36.

Salhi, S., M. Sari, D. Saidi and N. Touati (1992) Adaptation of some vehicle fleet mix heuristics, *Omega*, **20** (5-6) 653–660.

Savelsbergh, M. W. P. (1992) The vehicle routing problem with time windows: Minimizing route duration, *INFORMS Journal on Computing*, **4** (2) 146–154.

Solomon, M. M. (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, **35** (2) 254–265.

Subramanian, A., P. H. V. Penna, E. Uchoa and L. S. Ochi (2012) A hybrid algorithm for the heterogeneous fleet vehicle routing problem, *European Journal of Operational Research*, **221** (2) 285–295.

Taillard, E. D. (1993) Parallel iterative search methods for vehicle routing problems, *Networks*, **23**, 661–673.

Taillard, E. D. (1996) A heuristic column generation method for heterogeneous fleet VRP, *Publication CRT-96-13*.

Tarantilis, C. D., E. E. Zachariadis and C. T. Kiranoudis (2008) A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities., *INFORMS Journal on Computing*, **20** (1) 154–168.

Tavares, G., Z. Zsigraiova, V. Semiao and M. Carvalho (2009) Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling, *Waste Management*, **29**, 1176–1185.

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

Tung, D. V. and A. Pinnoi (2000) Vehicle routing–scheduling for waste collection in Hanoi, *European Journal of Operational Research*, **125** (3) 449–468.

Wang, X., C. Xu and H. Shang (2008) Multi-depot vehicle routing problem with time windows and multi-type vehicle number limits and its genetic algorithm, paper presented at the *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, 1–5.

Xu, H., Z.-L. Chen, S. Rajagopal and S. Arunapuram (2003) Solving a practical pickup and delivery problem, *Transportation Science*, **37** (3) 347–364.