A Particular Vehicle Routing Problem Arising in the Collection and Disposal of Special Waste

Roberto Aringhieri* Maurizio Bruglieri‡ Federico Malucelli‡
Maddalena Nonato§

*Dipartimento di Tecnologie dell'Informazione
Università di Milano
Crema, Italy
aringhieri@dti.unimi.it

‡Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza L. da Vinci 32, 20133 Milano, Italy
{bruglier,malucell}@elet.polimi.it

§Dipartimento di Ingegneria
Università di Ferrara
Via G. Saragat 1, 44100 Ferrara
mnonato@ing.unife.it

1 Introduction

The collection of bulky recyclable waste produced by households is organized by establishing several solid waste transfer stations (TS), so called *isole ecologiche* in Italy, spread in the territory nearby urban areas. Each TS hosts several containers, one for each type of collected waste material: e.g., paper, metals, trim yard, wood, glass, and so on. Residents bring their waste to the TS and dispose it into the appropriate container, according to the material. Containers are of different type, depending on their access side (left, right or rear access skips) or on the presence of a compacting equipment, and this may vary from TS to TS. Once a container is full, a disposal request is issued, consisting of the following two actions, to be carried out not necessarily in this order: i) the full container is brought to a disposal plant to be emptied, such as a landfill or a recycling facility, ii) an empty container of the same type is brought to the TS.

A fleet of homogeneous vehicles is available. Each vehicle can carry a single container at a time, either empty or full. In the common practice, the two actions a service request is made of are handled as a whole by a single vehicle, as described in [1]. On the other hand, splitting the two actions introduces a substantial degree of freedom, as it emerges in [3], since any sequence of operations is allowed and the loading and unloading of a container is not necessarily assigned to the same vehicle.

The problem addressed in this paper is the following. Consider a fleet of homogeneous vehicles hosted at a single depot and a set of additional empty containers of given types stored at the depot. Assume that the location of the available disposal plants for each type of material is given and all travel times along the road network are known. Given a set of service requests, the aim is to determine the vehicle routes starting and ending at the depot involving pick-up of full containers at TSs, dumping operations at appropriate disposal plants, delivery of empty containers where required, while minimizing both the number of vehicles and the global travelled time, subject to a maximum route duration constraint.

The problem can be modeled as a particular Asymmetric Vehicle Routing Problem (AVRP) on a suitable graph (see section 2) whose peculiar structure allows us to devise efficient heuristic algorithms (section 3). In a preliminary computational experience we compare the results of our algorithms with those currently implemented in some real life cases of a regional area in central Italy (section 4).

Closely related problems are addressed in [1], [3] and [5], but as far as we know the general case, dealing with multiple disposal facilities and limited number of available containers, has never been tackled. Indeed, [5] solve the version with unlimited containers, exploiting the tight bound provided by the solution of an associated transportation problem. However, such bound may turn rather loose in case of limited containers as containers become a common resource that has to be shared among all routes. In such a case, the decision of serving an empty request by bringing a new container from the depot or by using a container that has just been emptied is a global decision whose feasibility depends on the whole solution, even in the simplest case when the same container can be used at most once.

2 The graph structure and the AVRP formulation

For the sake of simplicity we consider a single depot.

Let $\{1, \ldots, n\}$ be the set of service requests. Each request i is characterized by a material μ_i, a container type β_i, and a TS γ_i. If two requests refer to the same container type, they are said to be *compatible*. The graph $G = (N, A)$ supporting our Vehicle Routing model does not directly map the physical network. Two main set of nodes, $E, F \subset N$ model service requests, in particular for each request i the two nodes f_i and e_i represent the full container to be brought to a specific disposal plant for material μ_i, and the request of an empty container of type β_i to replace the full one at γ_i, respectively. Moreover, we consider a node D representing the depot, K nodes $d'_1, \ldots, d'_K$ representing the possible pick-up of the K available empty containers located at the depot and K nodes $d''_1, \ldots, d''_K$ representing the return of empty containers of the same type to the depot.

Notice that we do not explicitly model nodes representing disposal plants, whereas disposal operations are embedded into some of the arcs of the graph. Two classes of arcs are given. Arcs in the first class model vehicles activities when loaded with a container, either full or empty, and are referred to as *loaded arcs*. The second class of arcs, so called *unloaded arcs*, models any vehicle trip when not carrying a container.

The main set of loaded arcs connect f_i and e_j for any pair of compatible requests (i, j). Arc (f_i, e_j) corresponds to picking up the full container of request i, taking it to and dumping it at the closest disposal plant for material μ_i on the way to the TS of request j, where the empty container is delivered. Note that, once the pair of requests is known, it is easy to determine the closest disposal plant and to evaluate the travel time which should include the time needed to carry out the loading-unloading operations as well as the dumping service. If $i = j$, then we have arc (f_i, e_i) corresponding to the simplest case of picking up a full container, emptying it at the closest disposal plant and bringing it back to its original site.

The main set of unloaded arcs concerns those connecting e_i to f_j for any pair of service requests (i, j). Arc (e_i, f_j) models a vehicle that has just delivered an empty container of type β_i at TS γ_i, and travels unloaded up to TS γ_j in order to pick-up the full container of request j. Note that requests i and j do not have to be compatible. The travel time related to this arc is simply the travel time from γ_i to γ_j, including container loading and unloading times. If $i = j$, arc (e_i, f_i) corresponds to bringing an empty container to satisfy request i and switch it with the full container.

Furthermore, since vehicles are supposed to leave the depot and return as unloaded, we consider as additional unloaded arcs all those: from D to each f_i node, from each e_i to the depot, from D to each $d'_1, \ldots, d'_K$, from each $d''_1, \ldots, d''_K$ to D; whereas additional loaded arcs are all those from each $d'_1, \ldots, d'_K$ to every compatible node e_i, and those all arcs from each f_i to every compatible node $d''_1, \ldots, d''_K$.

Remind that each node d'_k models the pick-up of an empty container available at the depot whereas d''_k models the delivery of the same container to depot. Thus arcs (D, d'_k), $k = 1, \ldots, K$ model the loading of a container on an unloaded vehicle at the beginning of a vehicle route; arcs (d''_k, D) , $k = 1, \ldots, K$ model the unloading of a container from a vehicle at the end of the vehicle route; whereas arcs (d''_k, d'_k), $k = 1, \ldots, K$ model the unloading of a container from a vehicle and the loading of another container at the depot during the vehicle route.

Observe that each arc corresponds to an operation that can be executed by a single vehicle. A *tour* is an elementary cycle through D in G, while a *feasible vehicle route* is a collection of tours disjoint on $N \setminus \{D\}$ having total duration less than or equal to a given L. A collection of feasible routes passing exactly once by $E \cup F$ and at most once by $d'_1, \ldots, d'_K$, $d''_1, \ldots, d''_K$ is a feasible solution of the waste disposal problem. Therefore solving the corresponding AVRP on G would solve our problem.

As the total travel time is the sum of the tours travel time, the number of vehicles is obtained by solving a bin packing problem with bin size L. As tours start and end at the depot, any permutation of the tours associated to the same bin yields a feasible route for that vehicle.

3 Heuristic algorithms

Any tour can be seen as an alternating sequence of e_i and f_j nodes, that is to say a sequence of loaded and unloaded arcs. Loaded arcs connect compatible nodes while unloaded arcs may connect nodes related to containers of any kind. Such particular graph topology suggests several neighborhood structures that can be exploited within a local search framework. We list them in the following, while computational results are discussed in section 4. Some neighborhoods can be seen as the specialization of classical VRP moves to G, and we refer to the terminology in [7, 2], while others have been suggested by the particular structure of G.

All the three neighborhoods aim at reducing the total route lengths with inter-route moves (neighborhoods N_1 and N_2) and intra-route moves (neighborhood N_3), however, N_1 and N_2 can be used with an appropriate move evaluation function to progressively shift nodes from a given route to others in order to reduce it to an empty route and save a vehicle. Indeed, while each single move is rather simple, they can be combined to give rise to quite sophisticated strategies. The only feasibility constraint to be checked in all such operations concerns maximum route duration and can be easily handled. Infeasible solutions could be considered by penalizing riding times exceeding the maximum, and not avoiding them.

Any move is subject to a vehicle cardinality feasibility check, that verifies the existence of a container packing solution with the current number of vehicles, with respect to an overworking parameter αL, $\alpha \in [0, 0.2]$ to handle temporarily infeasible solutions.

The starting solution is computed by applying a modified Clarke and Wright algorithm [4] which proved to be quite effective for this kind of problems as pointed out in [5].

3.1 Definition of neighborhoods

Neighborhood N_1: string exchange and relocation

Neighborhood N_1 is based on the exchange of two sequences s_1 and s_2 of consecutive arcs between a pair of tours. The sequences under consideration contain an odd number of arcs and start and end with unloaded arcs. By allowing at each step only pairs of sequences with $|s_1| = \sigma$ and $1 \leq |s_2| \leq \sigma$, for $\sigma = 1, \ldots, \sigma_{max}$ the neighborhood size can be controlled. Note that $|s_1| = 1$ has the effect of shifting a sequence from a tour to another one (i.e., relocation). Note that, since this neighborhood involves unloaded arcs only, compatibility does not have to be accounted for.

Neighborhood N_2: string cross

In this case we apply the classical string cross move to pairs of tours. The move is applied to either a pair of unloaded arcs or to a pair of compatible loaded arcs.

Neighborhood N_3: intra-tour string relocation

This move is a 3-opt move without arc reversing performed on a single tour restricted to unloaded arc triplets. This restriction allows us to neglect the effect of reversing a sequence containing loaded arcs.

Neighborhood N_4: intra-tour loaded arcs reversing

This move can be seen as a 2-opt move restricted to a sequence of compatible loaded arcs.

Neighborhood N_5: intra-tour additional empty container

This move consider the availability of an additional container of a given type. The use of the container is suitably inserted into a tour, thus involving the reversing of some loaded arcs. In practice this move is an extension of N_4.

Composite Neighborhoods

Families of moves are obtained suitably combining $N_1, \ldots, N_5$ in order to create sequences of compatible loaded arcs by string exchange, crossing and relocation and eventually reversing them.

4 Preliminary computational results

A preliminary computational campaign has been carried out on real data provided by the waste collection company operating in the regional area of Perugia. The company serves ten TSs distributed in an area of about 450 km^2, six different types of containers, three disposal centers used to recycle ten different types of material. The problem instances, deriving from the daily operations, involve three vehicles and up to 11 service requests, they require quite long hauls and they have a time limit of 375 minutes. Even though these instances are quite small, they provide useful indications. These instances are hardly solvable with commercial ILP solvers and the first results provided by the proposed heuristics favorably compare with the exact solution values. Moreover the resulting routes are extremely better than those operated by the company, saving travel time and in certain cases saving also one vehicle out of the three devoted to this kind of service. In the light of the obtained results, the company is thinking to extend the service also to industrial waste collection, which would significantly increase the size of the instances. Also extensions to multiperiod settings will be considered. Here we report some results on some real instances. We report the total travel time of the optimal solution (Opt), the percentage gap between the optimal solution and the modified Clarke and Wright (Gap CW), the percentage gap between the optimal solution and the local search (Gap LS), and the the percentage gap between the local search and the company solution (Gap G). As far as number of vehicles is concerned, entries in boldface report the cases in which the local search is able to spare a vehicle compared to the company solution. CPU times on a 2GHz PC are also reported.

References

[1] C. Archetti, M.G. Speranza, Collection of waste with single load trucks: a real case, `www.eco.unibs.it/dmq/speranza/`

[2] E. Aarts and J.K. Lenstra (eds.), Local Search in Combinatorial Optimization, John Wiley and Sons, Chichester, UK, 1997.

[3] L. Bodin, A. Mingozzi, R. Baldacci, M. Ball, "The Rollon–Rolloff Vehicle Routing Problem", Transportation Science 34 (3) 271-288 (2000).

[4] G.Clarke and J.V.Wright, "Scheduling of veichles from a central depot to a number of delivery points", Operations Research 12, 568–581 (1964)

Instance	Req	Opt	Gap CW %	Gap LS %	Gap G %	CPU Opt	CPU LS
17.11.03	8	668	7.04	**5.54**	11.91	3.52	0.03
18.11.03	3	300	14.00	0.00	0.00	0.1	0.05
19.11.03	7	615	0.00	0.00	0.00	3.44	0.04
20.11.03	8	701	21.83	0.71	15.16	7.56	0.05
21.11.03	6	626	**2.56**	**0.15**	3.00	0.1	0.01
22.11.03	9	843	15.18	15.18	3.09	21.32	0.03
24.11.03	8	684	24.12	**0.88**	3.19	3.05	0.06
25.11.03	8	575	2.61	2.61	13.90	19.42	0.07
26.11.03	6	599	1.17	1.17	12.05	6.71	0.03
27.11.03	8	839	9.30	8.82	6.79	5.78	0.01
28.11.03	6	606	2.97	2.97	12.02	0.1	0.01
29.11.03	11	882	2.72	1.59	19.98	0.1	0.07

Table 1: Some real case instances: travel times in minutes for each used vehicle

[5] L. De Muelemeester, G.Laporte, F.V.Louveaux, and F. Semet, “Optimal Sequencing of Skip Collections and Deliveries,” Journal of Operational Research Society 48, 57–64 (1997).

[6] P. Hansen and N. Mladenovic, ”Variable Neighborhood Search”, Computers and Operations Research 24, 1097–1100 (1997).

[7] P. Toth, D. Vigo (eds). The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. S.I.A.M., Philadelpia, PA, 2002.