
Tânia Rodrigues Pereira Ramos† Maria Isabel Gomes-Salema‡ Ana Paula Barbosa-Povoa*  
† ISCTE Business School, Lisboa {tania.ramos@iscte.pt}  
‡ Centro de Matemática e Aplicações, FCT-UNL, Monte da Caparica {mirg@fct.unl.pt}  
* Centro de Estudos de Gestão do IST, UTL, Lisboa {apovoa@ist.utl.pt}

Abstract

This paper aims to support tactical decisions in a recyclable waste collection system design through the definition of service areas in systems with more than one depot. Three types of recyclable materials are to be collected, so the problem under study is the multi-product, multi-depot vehicle routing problem. A mixed-integer linear programming model is formulated considering two alternatives constraints to eliminate subtours (Miller-Tucker-Zemlin and the Subtour Formulations). A heuristic approach to define service areas, based on the concept of borderline nodes, is also described and the solutions obtained are compared with the optimal solutions from the regular branch-and-bound method.

Keywords: Reverse logistics, Vehicle routing problem, MILP formulation, Heuristic

1 Introduction

In the recent years, reverse logistics has become one of the key functions in logistics systems, involving the flow of used products/materials from the point of consumption to the point of origin for the purpose of recapturing value or proper disposal (Rubio et al. 2008).

This work will focus on recyclable waste collection systems design. Considering waste packaging, there are usually three types of materials used in packaging that can be recycled: glass, paper and plastic/metal. The final consumer is responsible to separate these materials and drop them in special containers. Those materials are then collected in a regular basis and taken to a treatment plant by the company responsible for the recyclable waste collection system. The design of such systems involves strategic, tactical and operational decisions. This paper aims to support tactical decisions since it focuses on the definition of service areas in collection systems with more than one depot. In addition to the definition of the vehicle routes to collect the waste, it is also necessary to decide from which depot the collection is to be performed. This aspect adds one decision level to the classical Vehicle Routing Problem, where more than one product is to be collected in different routes. The resulted problem is modelled as a multi-product, multi-depot vehicle routing problem. To solve it, two approaches are presented: an exact and a heuristic method. The two methods are described and the results obtained for eight instances are compared. For the exact method, two formulations for the subtour elimination are also presented and compared.

This paper is structured as follows. After a brief review of the literature on multi-depot vehicle routing problem (MDVRP) in Section 2, we characterise generically the mathematical model in Section 3. In Section 4 we describe the heuristic approach developed. In Section 5 the computational results are presented and compared. Finally, we draw conclusions and discuss future work directions.
2 Literature Review

MDVRP is usually defined by a graph $G=(V, A)$, where $V$ is the vertex set and $A$ is the edge set. The vertex $V$ is partitioned into two subsets $V_d = \{v_1, \ldots, v_n\}$ and $V_c = \{v_{n+1}, \ldots, v_{n+1}\}$, representing the set of depots and the set of cities or clients. At each depot is based on a fixed schedule, and the routing information is fed back to the future assignments (Crevier et al. 2007).

The MDVRP consists of constructing a set of vehicle routes in such a way that: (1) each route starts and ends at the same depot, (2) each customer is visited exactly once by a vehicle, (3) the total demand of each route does not exceed the vehicle capacity, (4) the total duration of each route (including travel and service times) does not exceed a preset limit and (5) the total routing cost is minimized.

MDVRP appears as a generalization of the classical Capacitated Vehicle Routing Problem (CVRP). CVRP is a hard combinatorial problem, which can rarely be solved to optimality for problems with more than 50 nodes, and it is usually tackled by means of heuristics (Cordeau et al. 1997). However, due to recent developments in integer programming software systems, the ability to solve large-scale instances has been improved (see Atamturk and Savelsbergh 2005 for reviewing solvers state-of-the-art). Baldacci et al. (2007) describes the recent exact methods for the CVRP, and presents and compares the different mathematical formulations of CVRP used in the literature: the two and three index vehicle flow formulation; the two-commodity flow formulation and the set partitioning formulation. The exact methods of Fukasawa et al. (2006) and Baldacci et al. (2008) are presented with more detail and the computational results obtained are compared. The exact methods are applied over a set of instances with number of customers between 12 and 199 and number of vehicles up to 14.

For the MDVRP, there are still few exact algorithms in the literature. Laporte et al. (1984), as well as Laporte et al. (1988) have developed exact branch and bound algorithms for solving the symmetric and asymmetric version of the MDVRP, respectively. On the contrary, there are several heuristic algorithms developed to solve MDVRP. Tillman and Cain (1972) present a method of savings with a modified distance formula to include the existence of several depots. Golden, Magnanti and Nguyen (1977) have introduced the concept of borderline node, based on the calculation of the ratio $r(i)$ proposed by Gillet and Johnson (1976). Renaud et al.(1996) developed the algorithm FIND, based on a tabu search, to improve the initial solution found by applying the heuristic Improved Petal after assigning each client to the nearest depot. Salhi and Sari (1997) developed a heuristic for the multi-depot vehicle fleet mix problem (MDVFM), and have also tested it on the MDVRP. The objective of this heuristic is to simultaneously build a set of routes, and to determine the composition of the vehicle fleet at a minimum total cost (this cost includes the sum of the vehicle fixed cost and mileage cost). Lim and Wang (2005) proposed two solution methodologies to solve the multi-depot vehicle routing problem with fixed distribution of vehicles (MDVRPF): two-stage and one-stage approaches. The two-stage approach decomposes the MDVRPF into two independent subproblems, assignment and routing, and solves them separately. In the one-stage approach, assignment and routing are solved in an integrated way. This approach assigns customers to depots while producing routes, with the routing information is fed back to the future assignments. Crevier et al. (2007) study an extension of the MDVRP in which vehicles may be replenished at intermediate depots along their route (multi-depot vehicle routing problem with inter-depots routes). The authors propose a heuristic combining the adaptative memory principle and a tabu search method for the generation of a set of routes and integer programming in the execution of a set partitioning algorithm for the determination of least cost feasible rotations (the authors define rotation as the set of routes assigned to a vehicle).

The two methods (both exact and heuristic) are presented by Parthanadee and Logendran (2006). These authors developed a mixed-integer linear programming model to the multi-product, multi-depot periodic distribution problem. A technique to find the lower bound for this problem (the selective LP relaxation) is presented. The results obtained with the
regular branch-and-bound method, the selective LP relaxation and with three tabu search heuristics, also developed by these authors, are compared.

3 Mathematical formulation

The mathematical formulation of the present multi-product, multi-depot recyclable waste collection routing problem was based on the models presented by Golden et al. (1977), Cordeau et al. (1997), Lim and Wang (2005) and Parthanadee and Logendran (2006).

The model assumptions are:

1. A collection site aggregates one or more containers of one or more recyclable materials;
2. The quantity of each recyclable material to collect at each collection site is previously estimated;
3. The quantity of each recyclable material to collect at each collection site is less or equal to vehicle capacity for that material;
4. Each type of recyclable material must be collected separately;
5. The collection frequency of each collection site depends on the recyclable material.

There were defined the following sets and parameters:

Sets

\[ I = \text{a set of nodes } \{i = 1, \ldots, w\}; I = I_c \cup I_d \]

\[ I_d = \text{a set of depot nodes } \{i_d = 1, \ldots, n\} \]

\[ I_c = \text{a set of collection sites nodes } \{i_c = n+1, \ldots, w\} \]

\[ M = \text{a set of recyclable materials } \{m = 1, \ldots, M\} \]

\[ K = \text{a set of vehicles } \{k = 1, \ldots, K\}; K = K_1 \cup \ldots \cup K_n \]

\[ K_n = \text{a set of } k \text{ vehicles, which belong to depot } n \]

\[ U = \text{a set of trips that a vehicle can make in the timeframe} \]

Parameters

\[ d_{ij} = \text{distance between node } i \text{ and node } j \]

\[ Q_{km} = \text{capacity of vehicle } k \text{ to collect material } m \text{ (in weight)} \]

\[ f_{rm} = \text{collection frequency for the material } m \text{ in the timeframe} \]

\[ p_{im} = \text{amount of material } m \text{ to be collected at collection site } i \text{ (in weight)} \]

\[ c_{im} = \text{number of containers of material } m \text{ at collection site } i \]

\[ t_m = \text{time required to collect site } i \]

\[ r_{ij} = \text{time required to travel from node } i \text{ to node } j \]

\[ e_{\text{Max}} = \text{maximum time allowed for a trip/route} \]

\[ T = \text{number of days in the timeframe} \]

The decision variables are \( x_{ijmku} \) which represent the routing solution: \( x_{ijmku} = 1 \) if site \( j \) is visited immediately after site \( i \), to collect material \( m \), by vehicle \( k \), on its \( u \)th trip in the timeframe; 0 otherwise. The objective function of the model focuses on minimizing the total distance travelled to collect all recyclable materials at collection sites over the timeframe:

\[
\text{Min } \sum_{i \in I_d} \sum_{j \in I_c} \sum_{m \in M} \sum_{k \in K_n} \sum_{u \in U} d_{ij} f_{rm} x_{ijmku}
\]

(1)

where, as defined, \( d_{ij} \) is the distance between node \( i \) and node \( j \) and \( f_{rm} \) is collection frequency for the material \( m \) in the timeframe.

In terms of constraints, the model takes into account the classical routing restrictions which impose that (i) each collection site with material \( m \) is visited exactly once, (ii) a vehicle cannot leave and return to a depot other than its home depot, (iii) the vehicle capacity and the duration of each trip are not exceed and (iv) the route continuity. The
Subtour elimination constraints are implemented considering the Miller-Tucker-Zemlin Formulation, where \( o_i \) is an integer variable associated with each collection site \( i \). (2) or the Subtour Formulations (3):

\[
\sum_{i \in S} \sum_{j \in S} x_{ijm} \leq |S| - 1 \quad \forall m, \forall k, \forall u, S \subseteq I, |S| \geq 2
\]

For this specific problem, there is the need to add two new constraints: (v) one assures that at each collection site, all recyclable materials are collected from the same depot and (vi) the other guarantees that in each depot and over the timeframe the maximum time available is not exceeded.

4 Heuristic algorithm

The heuristic approach developed considers two objective functions to define the service areas: minimization of variable costs (function of the travelled distances by the collection vehicles) and minimization of the differences in workload between depots, according to the human resources available at each depot (promotion of equity between depots). However, the heuristic has parameters that allowed the decision maker to adjust the weight of each objective. Thus, and for the work here presented, only the objective that minimizes the total distance travelled is considered, and the results are compared with the ones of the mathematical formulation described above.

The definition of service areas consists in assigning the collection sites to the depots. This assignment always impacts on the quality of the VRP solution, since it influences and is influenced by the routing. Consequently, the heuristic approach aims at building service areas and defining routes, simultaneously.

As the collection sites are scattered over the territory and have a different relative positioning regarding the depots, the proposed heuristic uses different assignment criteria considering sites specific location. To differentiate locations, the concept of borderline node introduced by Golden et al. (1977) is used.

The first step of the heuristic is to classify collection sites as borderline and non-borderline nodes. Depending on this initial classification, the moment of assignment into depots and the assignment criterion to be used will be different, as will be specified ahead. The structure of the algorithm by modules is represented in Figure 1.

A brief description of each module is given below. For further details see Ramos and Oliveira (2009).

4.1 Classification of the collection sites as Borderline and Non-Borderline Nodes

The classification of the collection sites as borderline and non-borderline nodes is based on the calculation of ratio \( r(i) \) proposed by Gillet and Johnson (1976). After computing the distance from site \( i \) to the nearest depot, and the distance from site \( i \) to the second nearest depot, the ratio \( r(i) \), that relates these distances, is compared to a specific limit.
defined *a priori* - parameter $\delta$. If $r(i,j) < \delta$, then collection site $i$ is classified as non-borderline node; if $r(i,j) \geq \delta$, then collection site $i$ is classified as borderline node. Parameter $\delta$ can assume values between $[0, 1]$. In order to apply this heuristic to instances with more than two depots, it was necessary to introduce two modifications:

- $r_j(i)$: The index $j$ was introduced so that the ratio between the distance from collection site $i$ to the nearest depot and the distance from collection site $i$ to $j$ nearest depot ($j$ varies from second to $n^{th}$) can be calculated;
- $\delta_{dd}$: Instead of just one value, it was assumed that it may have different values for each pair of depots, which originated a matrix $\delta_{dd}$.

### 4.2 Assignment of the Non-Borderline Nodes to the Depots

The non-borderline nodes are much closer to one depot than to any other, so they will be served from its closest depot in order to minimize the distance travelled on the collection routes. However, that depot may not have available capacity (measured by the number of available vehicles and the number of available labour hours in the timeframe) to collect all the non-borderline nodes assigned to it. In these circumstances, the question of which nodes should be out of this initial assignment is posed. To answer it, an urgency assignment is calculated for each non-borderline node already assigned to the nearest depot. The urgency is a way to define a precedence relationship between collection sites. This precedence relationship determines the order in which collection sites must be assigned to depots (Giosa et al. 2002). Thus, for each depot there is a list of the non-borderline nodes assigned to it, ranked by urgency assignment. Whenever the depot does not have enough capacity to collect all the assigned nodes, the algorithm runs down this list and removes the nodes, starting by the last one in the list (the one with the least urgency value), and stops when the depot has capacity to collect all the remaining nodes.

To determine whether a depot has enough capacity to collect recyclable materials from all its non-borderline nodes, it is necessary to define routes and compute the time spent in the collection. Thus, the final decision on what nodes are assigned to each depot will only be made after designing the routes.

### 4.3 Assignment of the Borderline Nodes to the Depots

The borderline nodes are those which are in a situation of uncertainty regarding its assignment, given the proximity to more than one depot. These nodes are considered “borderline” between two or more depots. Therefore their assignment will always be conditioned by these depots. The decision to assign a borderline node to a depot is based on an indicator that measures the attractiveness of each depot. This attractiveness measure ($M_{id}$) indicates which depot is more attractive for each borderline node, considering the workload of depot $d$ (number of working hours to collect the nodes already assigned to depot $d$) and the distance between node $i$ and the nearest route belonging to depot $d$ (since there are provisional routes already defined that may be expanded to include the borderline nodes). The attractiveness measure $M_{id}$ of depot $d$ to node $i$ is numerically defined by the linear relation

$$M_{id} = 1 - \left[ \alpha \frac{\text{dis}_{ir}}{\sum_{r \in D_i} \text{dis}_{ir}} + (1 - \alpha) \frac{\text{CT}_{d}/K_d}{\sum_{d \in D} \text{CT}_{d}/K_d} \right]$$

where $D_i$ is the set of depots whose node $i$ is considered “borderline”, $\text{dis}_{ir}$ is the distance between node $i$ and the nearest route $r$ belonging to depot $d$, $\text{CT}_{d}$ is the workload of depot $d$, $K_d$ is the number of vehicles at depot $d$ and $\alpha$ is the weight of the minimize distance objective.
4.4 Routing

The route definition appears in this algorithm as a secondary element albeit necessary to reach the main goal: defining balanced service areas by promoting the minimization of the distance on the collection routes of the nodes belonging to those areas. Therefore, its definition is achieved through the use of a simple and effective heuristic, such as the Clarke-Wright algorithm (Clarke and Wright (1964)).

The routes are defined by recyclable waste type and by depot. The routes are repeated along the timeframe according to the collection frequency for each waste type (the collection frequency is the same for each material and it is a given parameter).

5 Determination of the optimal solution

In this section, we evaluated the efficiency of the proposed heuristic algorithm by applying it to a set of test instances and comparing some performance measures (i.e., the quality of solutions and computational time) with the branch-and-bound method. Because the multi-product, multi-depot vehicle routing problem is a very complex mixed-integer problem, the instances that can be solved to optimality will typically be small. Therefore, eight small instances were generated. The structure of these instances is presented in Table 1.

Table 1: Structure of the problem instances

<table>
<thead>
<tr>
<th>Problem</th>
<th>No. of Depots</th>
<th>No. of Collection Sites</th>
<th>No. of recyclable materials</th>
<th>No. of vehicles</th>
<th>No. of trips</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>11</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>12</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>17</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

The instances are solved using the branch-and-bound method implemented in the solver of the CPLEX Optimizer 11.1.1. A Intel Core 2 Quad CPU Q6600 @ 2.40 GHz is used. In this testing, the branch-and-bound computation time is arbitrarily limited to 8 hours.

In this work, we also intend to compare the performance of the Miller-Tucker-Zemlin Formulation with the Subtour Formulation (see equation (2) and (3) respectively, at Section 3). Table 2 compares the total number of constraints and variables for the model described in Section 3, considering both alternatives for subtour elimination.

Table 2: Comparison of the number of constraints and variables on the two subtour formulations

<table>
<thead>
<tr>
<th>Problem</th>
<th>MZT Formulation</th>
<th>Subtour Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of constraints</td>
<td>Number of variables</td>
</tr>
<tr>
<td></td>
<td>Discrete Total</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>132 188 194</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1458 672 1773</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1848 2376 2548</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2155 2376 2567</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2553 2808 3018</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2015 2520 2677</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14521 17433 17883</td>
<td></td>
</tr>
</tbody>
</table>

To prevent subtours, it was enough to compute the family of subtours constraints with 2 and 3 connections.

The heuristic algorithm was applied to the test problems listed in Table 1. The heuristic was written in MATLAB and run on a Genuine Intel(R) CPU T2130@1.86 GHz.
Since the heuristic algorithm has two objectives and the exact method only considers distance minimization, the parameters in the heuristic were adjusted to accomplish this situation. Thus, parameter $\delta_{d}$ was set to value 0.5, which generated some borderline nodes and the parameter $\alpha$ was set to value 1, which means that the algorithm only takes into account the distance objective.

The results from both the branch-and-bound search and the heuristic algorithm are presented in Table 3.

Table 3: Comparisons between the optimal values and the heuristic solution

<table>
<thead>
<tr>
<th>Problem</th>
<th>CPLEX @ MTZ Formulation</th>
<th>CPLEX @ Subtour Formulation</th>
<th>Heuristic Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90*</td>
<td>90*</td>
<td>90*</td>
</tr>
<tr>
<td>OFV</td>
<td>0,11</td>
<td>0,11</td>
<td>0,06</td>
</tr>
<tr>
<td>Time (s)</td>
<td></td>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>Dev.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>527*</td>
<td>527*</td>
<td>542</td>
</tr>
<tr>
<td>OFV</td>
<td>4</td>
<td>0,5</td>
<td>0,06</td>
</tr>
<tr>
<td>Time (s)</td>
<td></td>
<td></td>
<td>2,8%</td>
</tr>
<tr>
<td>Dev.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>791*</td>
<td>791*</td>
<td>801</td>
</tr>
<tr>
<td>OFV</td>
<td>104</td>
<td>44</td>
<td>0,18</td>
</tr>
<tr>
<td>Time (s)</td>
<td></td>
<td></td>
<td>1,3%</td>
</tr>
<tr>
<td>Dev.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>694*</td>
<td>694*</td>
<td>752</td>
</tr>
<tr>
<td>OFV</td>
<td>100</td>
<td>27</td>
<td>0,18</td>
</tr>
<tr>
<td>Time (s)</td>
<td></td>
<td></td>
<td>8,4%</td>
</tr>
<tr>
<td>Dev.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OFV is the objective function value. The * indicates the optimal solution. The OFV without * indicates the best integer solution found. Time is the time taken to compute the solution (in seconds); the “-” implies that CPLEX was terminated after 8 hours (28800 seconds). Dev. = (OFV$_{heuristic}$/Best OFV – 1)×100.

Using regular branch-and-bound, the optimal solutions are found in all test instances, except for problem 8. The Subtour Formulation proved to be more efficient in the smallest instances since the computational time to compute the optimal solution is shorter than the MTZ Formulation one. The Subtour Formulation produces a large number of constraints (see Table 2) but it reveals to be more efficient in the search for the optimal solution in the smallest instances (instance 1 to 5). For instance 6, the Subtour Formulation failed to prove the optimal solution within the 8-hour time limit. Both MTZ Formulation and Subtour Formulation failed to report the optimal solution for instance 8 within the limit of 8 hours.

Even though, the heuristic algorithm provides “good” solutions since the percentage deviation never overcome 11% (the average deviations is 3,6%). The heuristic algorithm only identifies the optimal solution for problem 1. The heuristic percentage deviation does not increase with growing of problems’ size as we can see in Table 3. In all test instances, the heuristic algorithm identifies a good solution in less than 1 second.

6 Conclusions

The multi-product, multi-depot vehicle routing problem is a very hard combinatorial problem, which have been tackled by heuristic algorithms. Due to the recent improvements on commercial optimizers, it is now possible to solve hard combinatorial problems by exact algorithms.

In this work we compare the performance of two alternatives constraints to eliminate subtours and compare their performance with a heuristic approach. In order to do so, eight small instances are generated, representing a multi-product, multi-depot vehicle routing problem.
The Subtour Formulation proves to be more efficient for the smallest test instances since the computational time to compute the optimal solution is shorter when compared with the MTZ Formulation. We also conclude that there is no need to formulate all the family of subtours (with c connections, where c is the number of collection sites) to prevent subtour formation in the solution. For larger scale problems, the MZT Formulation shows to be more efficient.

The heuristic algorithm provides “good” quality solutions in less time than branch-and-bound method.

This work represents the first step to solve a large size real-case in a recyclable waste collection system with five depots and 212 collection sites. The next steps will consist in modifying the objective function in the mathematical model to comprise the minimization of the differences in workload between depots and the use of other formulations in order to solve large-scale instances.

7 References


