A Decision Support System for a Real Vehicle Routing Problem

Rubén Ruiz, Concepción Maroto, Javier Alcaraz

Dpto. de Estadística e Investigación Operativa Aplicadas y Calidad.
Universidad Politécnica de Valencia
Camino de Vera S/N, 46021, Valencia, SPAIN

{rruiz,cmaroto,jalcaraz}@eio.upv.es

28th January 2003

Abstract

The Vehicle Routing Problem has been widely studied in the literature, mainly because of the real world logistics and transportation problems related to it. In the present paper, a new two-stage exact approach for solving a real problem is shown, along with decision making software. In the first stage, all the feasible routes are generated by means of an implicit enumeration algorithm; afterwards, an integer programming model is designed to select in the second stage the optimum routes from the set of feasible routes.
The integer model uses a number of 0-1 variables ranging from 2,000 to 15,000 and gives optimum solutions in an average time of 60 seconds (for instances up to 60 clients). An interactive Decision Support System was also developed. The system was tested with a set of real instances and, in a worst-case scenario (up to 60 clients), the routes obtained ranged from a 7% to 12% reduction in the distance travelled and from a 9% to 11% reduction in operational costs.

**Keywords:** Decision Support Systems, Logistics, Routing, Transportation.

## 1 Introduction: The Problem

Logistics and transportation account for a large portion of the economies of the developed countries. Therefore, governments and private companies focus their attention on developing systems that could aid logistics managers to lower costs and achieve greater flexibility.

A lot of research has been performed in the field of logistics and many techniques have been already developed, from the Travelling Salesman Problem to complex dynamic routing problems. One of the prominent problems in logistics is the Vehicle Routing Problem or VRP. This problem can be briefly described as a set of $N$ clients or customers with known and deterministic demands $d_i$, $i \in 1, \ldots, N$ that have to be served from a central depot or origin with a fleet of $t$ delivery trucks of known capacity $Q$. Normally, the objective is to minimize the total distance travelled by the truck fleet, but it is also common to minimize route costs.
This combinatorial problem in its simplest form, is \( \mathcal{NP} \)-Hard (see Garey and Johnson, 1979), and was initially studied by Dantzig and Ramser (1959) and by Clarke and Wright (1964). A much more comprehensive and detailed study of this problem can be found in Bramel and Simchi-Levi (1999), Crainic and Laporte (1999) and Ball et al. (1995).

This paper deals with a real problem in close relation to the VRP. Below, follows a complete explanation of this problem. Nanta S.A. is the leading Iberian feed compounding, with production plants in Spain and, more recently, in Portugal. It offers extensive ranges of pig and poultry feeding, as well as feed for ruminants, rabbits and other livestock, all of which amount to more than 150 different products. It currently operates a total of 11 factories, ten of them in Spain and one in Portugal. The plant under study is located in Valencia, at the east coast of Spain. Nanta’s plants are highly automated and the productive process is computerized. With the production already solved, the company’s factories and in particular, the plant located in Valencia, face other important issues, such as feed compound distribution to clients. Every day, more than 800,000 Kg. of compound feed have to be distributed to an average of 70 clients, out of a total of 700 different clients located at distances ranging from 15 to 450 Km. away from the production plant. The orders placed have to be delivered with an outsourced truck fleet the following day. Furthermore, between 5 and 30 urgent orders are additionally placed each day, causing a certain degree of havoc both at production and logistic levels. The existence of rush (urgent) orders is a very problematic issue, as rush orders usually
have to be delivered by single-stop trucks due to the impossibility of re-organizing all the routes in such a short period of time.

Each truck has its unique characteristics and capacities, the capacities ranging from 12 Tn. to 24 Tn. Every truck has a number of watertight compartments used to carry different kinds of feed compound, each compartment has a capacity of 4 Tn. Three different kind of trucks are considered (this is a simplification that does not affect the solution). These trucks are the trailers, with 6 compartments (24 Tn.), 4 axle trucks that also have 6 compartments but are smaller, and the 3 axle trucks, which only have 4 compartments (16 Tn.). As we will see, it is very important to remark that from one compartment, only one client can be served. This is because the trucks do not include a method for measuring how much feed compound is being poured to each determinate client, and instead, each compartment is assigned to a client order, therefore, the number of clients served will depend on the number of compartments. For example, a trailer can visit only 6 clients at most.

The company uses a complex function when evaluating the cost of a given route, which can be explained as follows: The company pays to the outsourced truck company a given amount of money depending on the distance travelled to the final stop of the truck. With that distance, a corresponding price per Tn. of load is looked up in a table, and then multiplied by the number of Tn. initially carried by the truck, which finally gives the amount to be paid.

There are some minimum loadings that have to be taken into account, the minimum load for a 24 Tn. truck is 23 Tn. and for a 16 Tn. is 15 Tn. so, if a truck
travels half empty, the company would have to pay the minimum load, for example, if a 24 Tn. truck carries 20 Tn. the company would have to pay for 23 Tn. thus paying for 3 Tn. of “air”.

In addition to this, a lot of constraints have to be considered; the maximum allowed weight for the trucks cannot exceed local regulations, and some trucks cannot reach some clients or pass along some roads, bridges or tunnels because of their size and/or weight. Also, the company sets a 450 Km. maximum distance for trucks to travel in single trips, and a minimum order weight of 4 Tn. (the capacity of a truck compartment).

Route building processes are not computerized and can be briefly explained as follows:

- First, the orders that would fill a whole truck are assigned to single-stop routes, for example, if a client demands 24 Tn. of feed compound, a single trailer would be used.

- The remaining orders are grouped and assigned to some predetermined geographical areas. Afterwards, a simple Geographical Information System (GIS) is used to manually generate routes in an attempt to minimize distances and use all truck capacity.

- The orders that do not fit into any of the routes previously generated are delivered with smaller trucks (16 Tn.) in single-stop routes. For orders larger than 16 Tn. the client is contacted in order to rearrange another delivery at a later time and/or another feed compound quantity (usually smaller so the
order can fit into an already built route).

The whole process can take as much as 5 working hours and can lead to very unsatisfying results, since many times clients are not served within datelines required or are served a lesser quantity of the product.

Many papers in the literature deal with the Vehicle Routing problem and its variations, some deal with traditional techniques such as integer programming and some suggest more recent techniques such as Genetic Algorithms or Tabu Search. Generally, the solution techniques used for solving the VRP can be separated into exact methods and heuristics or, more recently, metaheuristics.

Regarding exact approaches, Önal et al. (1996), proposed a mixed integer programming model and a dynamic programming method which was applied to a real problem of fuel dispatching in an agribusiness firm where multiple deliveries are considered. Both methods are compared with favourable results for the dynamic approach. However, the problem instances tested range from 5 to 9 clients. The real problem illustrated in this paper considers a larger number of 70 clients. Blasum and Hochstädtler (2000) show an interesting modification of the classical Branch and Cut algorithm, however the algorithm proposed only offers a solution for a simple variation of the VRP, the symmetric Capacitated Routing Problem (CVRP). Blasum and Hochstädtler manage to solve some synthetic instances to optimality but at a rather high computational cost.

Toth and Vigo (2002) present a review of models and exact techniques for the VRP, in which several branch and bound approaches for the VRP are reviewed
and evaluated. Although some of the methods are suitable for fairly-sized instances, to the best of our knowledge, none of them deal with problems other than the basic version of the VRP (only vehicle capacity is considered).

As we can see, there are some serious drawbacks regarding exact approaches. Many times exact algorithms are not suitable for real instances since the computational times needed in order to obtain a solution are unviable and the VRP variations considered are far too simple for the methods to be of any practical use. Furthermore, it is not always easy to adapt integer programming models or Branch and Bound algorithms to more complicated VRP variations.

Heuristic approaches are used when finding an optimal solution for bigger instances in a reasonable amount of time is not deemed possible. There are many heuristic and metaheuristic methods available. Bachem et al. (1996) apply a general improvement heuristic, called Simulated Trading in a complex parallel computing environment with promising results.

Xu and Kelly (1996) use a Tabu Search algorithm to solve the standard VRP. Another metaheuristic applied to the VRP is given in the work of Bullnheimer et al. (1997). In this case an Ant System is implemented but only for a very simplistic VRP with an homogeneous vehicle fleet. Rego (1998) also considers a Tabu Search algorithm for the VRP, but again, only capacity and route length restrictions are considered in the problem.

De Backer et al. (2000) present an interesting work with a Constraint Programming paradigm (using ILOG Solver) and two metaheuristics, Tabu Search and
Guided Local Search. The results are compared against Solomon’s classic benchmark (1987). In the work of Van Breedam (2001) 3 different metaheuristics for the classic VRP are compared. These are Descent Search, Simulated Annealing and Tabu Search. In a later work (Van Breedam, 2002) the author compares a total of 10 heuristics, in this case, slightly more sophisticated variations of the VRP are considered.

More recently, Ralphs et al. (2002), proposed new techniques to solve the VRP with a fleet of vehicles of uniform capacity. The method used is a separation algorithm with a parallel branch and cut framework, called SYMPHONY.

Although heuristic and/or metaheuristic approaches can solve larger VRP instances, we still find that real VRP problems with more realistic constraints are seldom considered. Moreover, there are few papers dealing with realistic problems. Önal et al. (1996) considered multiple visits and pickup routes. Tarantilis and Kiranoudis (2002) developed a complex Spatial Decision Support System (DSS) for the VRP in which GIS are used. Lastly, Angelelli and Speranza (2002) solved a wastage pickup problem. Once again, none of this papers deals with the specific constraints of our problem and furthermore, the methods proposed are not easy to adapt.

We completely agree with Desrochers et al. (1999) when they remarked that the application of the techniques appeared in the literature required a great deal of knowledge and expertise because of the existing gap between the problems
considered in the literature and the real problems found in practice. They proposed a model and an algorithm management system to provide support in the modelling of more realistic logistic problems. Implantation of all aforementioned techniques in plants and factories is not so straightforward. As we have seen, many of the techniques do not provide the flexibility and responsiveness that a real logistic environment needs. Many of the cited references deal with simplistic versions of the VRP, which do not take into account heterogeneous vehicle fleets or other optimization criteria different from total route length minimization, not to mention specific constraints related to Nanta’s specific problems such as the limited number of stops per route, the inability of some trucks to reach some clients or the minimum loadings penalties for the trucks.

The main objective of this work is to develop a simple method for solving the real logistic problem considered, which is the daily dispatching of feed compounds to an average of 70 clients. Additionally, this paper introduces a tool for decision making that closes the gap between the method proposed and the satisfactory daily use at the plant. The remaining of the paper is organized as follows: In section 2.1, a first attempt to solve the problem with traditional modelling techniques is shown. Section 2.2 explains the two-stage approach for solving the problem. In the first phase, an implicit enumeration algorithm generates every feasible route for the problem. In the second phase, an integer programming model chooses the best set of routes from the feasible route set. In section 3, a complete description of the developed decision support system is given, along with some
results and tests. Finally, in section 4 some conclusions on the study are provided.

2 The Model

2.1 A vehicle routing problem approach

Given the previous problem statement, an initial approach used for solving the problem was to use a simple integer programming model. The model used for solving the problem is explained below:

Parameters:

\[ Q \] = capacity of the vehicle
\[ N \] = number of clients, or truck stops
\[ d_i \] = demand of client \( i, i > 0 \)
\[ C_{ij} \] = distance between client \( i \) and client \( j \)

Variables:

\[ X_{ij,i\neq j} = \begin{cases} 
1, & \text{if a truck goes from client } i \text{ to client } j \\
0, & \text{otherwise} 
\end{cases} \]

where \( i, j \in \{0, \ldots, N\} \), being 0 the origin or depot
Objective function:

\[
\text{Min} \sum_{i=0}^{N} \sum_{j=0,j \neq i}^{N} C_{ij} X_{ij}
\]

Constraints:

\[
\sum_{i=1,i \neq j}^{N} X_{ij} = 1, \forall j, j \in \{1, \ldots, N\} \tag{1}
\]

\[
\sum_{j=1,j \neq i}^{N} X_{ij} = 1, \forall i, i \in \{1, \ldots, N\} \tag{2}
\]

\[
\sum_{i=0}^{N} \sum_{j=0,j \in S}^{N} X_{ij} \leq |S| - 1 \tag{3}
\]

\[
\sum_{i=0}^{N} \sum_{j=0,j \in T}^{N} X_{ij} \leq |T| - k \tag{4}
\]

We can see that the objective function is minimizing the total distance travelled. Constraints (1) and (2) ensure that every client is visited by a truck and that every truck leaves each client. Constraint (3) eliminates subtours, i.e. tours that do not start and finish at the depot, this constraint is added for every possible subset \(S\) of clients, not including the depot. Finally, constraint (4) takes into account the load of the trucks, disallowing overloads; this constraint is added for every set of clients \(T\), including the depot, which constitutes more than a possible truckload (every set that satisfies: \(\sum_{i \in T} d_i > Q\)) and \(k\) is the minimum number of clients that have to be taken from \(T\) to avoid overloading.

The model does not take into account all the aspects of Nanta’s problem. We tried this model first, as an initial attempt to evaluate the following possibilities.
The model was implemented and solved with LINGO v6 (Schrage, 1998). The model proposed proved to be inappropriate for many reasons, some of which can be resumed as follows:

- **Inefficiency**: The model uses $N^2$ variables and needs an unacceptable amount of computer time in order to find an optimal solution, as a matter of fact, none of the instances tested could be adequately solved. Table 1 shows five test instances, which are five simplified orders sets, taken from factory working days. As we can see, we stopped the model after 7,200 seconds without having reached an optimal solution.

- **No Complete**: The model does not take into account all the significant aspects of the problem, and this situation affects the measure of effectiveness. In particular, some restrictions of the problem are not satisfied, such as the different truck sizes and their inability to reach some clients.

We could have used a more sophisticated model in order to better control trucks and routes. In Thangiah (1995), a much more detailed model is explained. In this model, trucks are linked to clients by binary variables and the transitions between clients are also linked to the trucks, therefore, the model requires $(N \cdot K) + (N^2 \cdot K)$ variables, being $K$ the number of available vehicles. For example, this model would require as much as 49,500 binary variables in order to serve 44 clients with 25 trucks and, as we have seen, this would have been really hard to solve in an acceptable amount of time.
Objective function: Minimizing the total distance travelled is not exactly the goal, we need to minimize the real cost of the routes. This is one of the major drawbacks of the model since, as we have seen in section 1, there is a complex route cost function. Minimizing the total distance travelled leads to many short routes with half-empty trucks, which are expensive for the company. Once again, we would have needed a more sophisticated model in order to minimize the real cost of the routes.

While attempting to solve the model’s inefficiency we tried a set partitioning of the problem. We partitioned the original instances by assigning the different clients to 3 big regions and 10 smaller zones, depending on the client’s location. With this approach, we can create smaller subproblems by dynamically solving regions or sets of zones in which there are fewer clients to be served and, therefore, less binary variables. This partition proved to be feasible in terms of computing time. Table 2 shows five simplified instances that were partitioned into three different

Table 1: Results for the initial model.

<table>
<thead>
<tr>
<th>Instance</th>
<th>clients</th>
<th>variables(^1)</th>
<th>constraints</th>
<th>Best solution found (km.)</th>
<th>computing time (sec.)(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>210999</td>
<td>35</td>
<td>989</td>
<td>1,405</td>
<td>2,528</td>
<td>7,200</td>
</tr>
<tr>
<td>220999</td>
<td>34</td>
<td>808</td>
<td>1,331</td>
<td>2,291</td>
<td>7,200</td>
</tr>
<tr>
<td>100599</td>
<td>31</td>
<td>791</td>
<td>1,121</td>
<td>1,843</td>
<td>7,200</td>
</tr>
<tr>
<td>121199</td>
<td>44</td>
<td>1,221</td>
<td>1,652</td>
<td>2,976</td>
<td>7,200</td>
</tr>
<tr>
<td>301199</td>
<td>38</td>
<td>1,086</td>
<td>1,565</td>
<td>2,549</td>
<td>7,200</td>
</tr>
</tbody>
</table>

\(^1\)The number of variables does not exactly match \(N\), this is because in the LINGO implementation of the model, many variables with known values are not declared, for example, it is obvious that \(X_{ij} = 0\) when \(i = j, i, j \in \{1, \ldots, C\}\).

\(^2\)A PC computer with an AMD Athlon processor running at 1200Mhz and a main memory size of 256Mbytes was used for all the tests in this paper.
sets. All five instances were solved to optimality, always under 5 minutes of computing time, which is a great achievement compared to the non-partitioned model.

However, this partition introduced some degree of complexity and a significant degradation of the solution. Each region or zone is solved independently from the adjacent regions, which can lead to many short routes running along the edges of zones. These short routes could be coalesced into a single one, probably resulting in cost savings for the company. This situation is depicted in Figures 1 and 2.
<table>
<thead>
<tr>
<th>Instance</th>
<th>partition</th>
<th>clients</th>
<th>variables</th>
<th>constraints</th>
<th>Objective (km.)</th>
<th>computing time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>210999</td>
<td>Region 1</td>
<td>10</td>
<td>98</td>
<td>155</td>
<td>657</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Region 2</td>
<td>13</td>
<td>145</td>
<td>237</td>
<td>763</td>
<td>38.00</td>
</tr>
<tr>
<td></td>
<td>Region 3</td>
<td>12</td>
<td>142</td>
<td>209</td>
<td>758</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>35</td>
<td>385</td>
<td>601</td>
<td>2,178</td>
<td>39.01</td>
</tr>
<tr>
<td>220999</td>
<td>Zones 1-4</td>
<td>10</td>
<td>72</td>
<td>155</td>
<td>321</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Zone 5</td>
<td>11</td>
<td>99</td>
<td>181</td>
<td>770</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Zones 6-10</td>
<td>13</td>
<td>167</td>
<td>239</td>
<td>1,126</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>34</td>
<td>338</td>
<td>575</td>
<td>2,217</td>
<td>5.02</td>
</tr>
<tr>
<td>100599</td>
<td>Region 1</td>
<td>9</td>
<td>73</td>
<td>131</td>
<td>437</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>Region 2</td>
<td>15</td>
<td>203</td>
<td>305</td>
<td>702</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>Region 3</td>
<td>7</td>
<td>63</td>
<td>89</td>
<td>415</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>31</td>
<td>339</td>
<td>525</td>
<td>1,554</td>
<td>100.01</td>
</tr>
<tr>
<td>121199</td>
<td>Region 1</td>
<td>13</td>
<td>154</td>
<td>221</td>
<td>804</td>
<td>35.00</td>
</tr>
<tr>
<td></td>
<td>Region 2</td>
<td>16</td>
<td>224</td>
<td>348</td>
<td>1,201</td>
<td>114.00</td>
</tr>
<tr>
<td></td>
<td>Region 3</td>
<td>15</td>
<td>196</td>
<td>273</td>
<td>827</td>
<td>102.00</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>44</td>
<td>574</td>
<td>842</td>
<td>2,832</td>
<td>251.00</td>
</tr>
<tr>
<td>301199</td>
<td>Zones 1-4</td>
<td>11</td>
<td>106</td>
<td>188</td>
<td>774</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Zones 5-6</td>
<td>12</td>
<td>138</td>
<td>196</td>
<td>846</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>Zones 7-10</td>
<td>15</td>
<td>196</td>
<td>273</td>
<td>827</td>
<td>110.00</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>38</td>
<td>440</td>
<td>657</td>
<td>2,447</td>
<td>114.01</td>
</tr>
</tbody>
</table>

Table 2: Results after the set partitioning.

2.2 Implicit enumeration and a revised model

As an alternative, we thought that an implicit enumeration of the routes would lead to a new model with much fewer variables that could be solved in a reasonable amount of time.

At a first glance, enumerating the routes might seem an inappropriate approach,
as the following example illustrates:

A truck cannot visit more than 6 clients, since it cannot carry more than 6 different kinds of feed compound and the minimum order weight is 4 Tn. hence, the number of possible routes for 50 clients and 6 stops per route is:

\[ V_{50,6} + V_{50,5} + V_{50,4} + V_{50,3} + V_{50,2} + V_{50,1} = 11,701,202,500 \]

Which accounts for the possible route variations of 6 stops, 5 stops, and so on. Obviously, this is a very big number. However, only a small fraction of the possible routes are actually feasible. For example, a possible route can be one visiting six clients located very far from each other and each one demanding 24 Tn. of feed compound. The resulting route would require a truck carrying 144 Tn. of product and probably travelling for more than 700 Km. This is clearly an infeasible route.

A close examination of the real data leads to the following conclusions:

- The number of routes with six clients is very low, not every route can have six stops, since the maximum allowed load of the trucks is limited; and, as we have seen, we can only serve one client from one compartment and the number of compartments is fixed in each truck.

- A route can not be longer than 450 Km. which also limits the number of routes with many stops.

- Some trucks can not be used to serve some clients due to constraints in weight, small roads, bridges or tunnels.
Taking into account these constraints, we found that for the previous example, the actual number of feasible routes was only 2,504.

A graphical example can be seen as follows:

Let us consider only three clients, named A, B and C. The feed compound requirements are 24, 16 and 8 Tn. respectively. The distance matrix between these three clients and Nanta can be seen in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>Nanta</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nanta</td>
<td>0</td>
<td>50</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>A</td>
<td>50</td>
<td>0</td>
<td>100</td>
<td>125</td>
</tr>
<tr>
<td>B</td>
<td>60</td>
<td>100</td>
<td>0</td>
<td>210</td>
</tr>
<tr>
<td>C</td>
<td>70</td>
<td>125</td>
<td>235</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Distance matrix for the simplified example.

Now let us consider every possible route for these clients. We enumerate all the possibilities and, for each resulting route, we calculate the kilometers travelled and the quantity of feed compound that the corresponding truck would need to carry. The resulting routes can be seen in Figure 3.

As we can see, we have a maximum of three stops per route, since we only have three clients. If we look at the enumerated routes, we find that not every one is feasible, for example, the route Nanta-A-B-Nanta is 210 Km. long, which is feasible, but it requires a truck carrying 40 Tn. of compound feed, which is not possible, and therefore, that route is infeasible.

By applying the same rule to all possible routes, we can find that the feasible routes are only 5, as depicted in Figure 4.

As a result, we developed an algorithm that, given a set of orders, enumerates
all the different routes. In this case, we could have encountered a problem, since enumerating all the possible routes would require an unacceptable amount of processing time due to the large number of them. However, we developed an implicit enumeration algorithm, which is a systematic evaluation of all possible solutions without explicitly evaluating all of them (see Winston, 1997 and Hillier and Lieberman, 2001). For example, let us consider again the previous example; We take client B, then one feasible route is Nanta-B-Nanta, travelling for 120 kilometers and carrying 16 Tn. of feed compound, then we take that generated route and see if we can generate more feasible routes by adding more stops. From that client B, we cannot go to client A, for the resulting route (Nanta-B-A-Nanta) would carry 40 Tn. of product and this would not be feasible. So, the Nanta-B-A-Nanta route is rejected and therefore, we do not add more stops to this route since it is not generated. This methodology ensures that the tree depicted in Figure 3 is adequately pruned.

The implicit enumeration algorithm gives, for each feasible route, the set of clients it visits, the total distance travelled, the total cost of the route, and the kind of truck to be used.

Once all feasible routes are generated, we need a model able to choose the best set of routes from the feasible route set; within this best set, every client has to be visited exactly once. The objective is minimizing the total cost (not the total distance travelled) of the routes, which is now known. This model is explained below:
Parameters:

\[ N = \text{number of feasible routes}. \]

\[ C = \text{number of orders}. \]

\[ PP_{nc} = \begin{cases} 
1, & \text{if the route } n \text{ visits client } c \\
0, & \text{otherwise} 
\end{cases} \]

\[ Cost_n = \text{Cost of the route } n. \]

where \( n \in \{1, \ldots, N\}, c \in \{1, \ldots, C\} \)

Variables:

\[ X_n = \begin{cases} 
1, & \text{if the route } n \text{ is done} \\
0, & \text{otherwise} 
\end{cases} \]

Objective Function:

\[ \text{Min} \sum_{n=1}^{N} Cost_n X_n \]

Constraints:

\[ \sum_{n=1}^{N} PP_{ni} X_n = 1, \forall i, i \in \{1, \ldots, C\} \]

All parameters are calculated by the implicit enumeration algorithm, so any changes in specification could be done in the algorithm without needing to change the
model.
This model proved to be extremely fast and robust, the solution times were always under 1 minute for normal tests, and under 5 minutes for models with over 35,000 binary variables. We tested the various options that LINGO offers for solving models, and found that the use of the different constraint cuts reduced the computing time; using Gomory and GUB cuts proved to be the best solution for all cases. It is also worth noticing that the model provides the optimum set of routes, i.e. there is no better way of delivering the compound feed under cost consideration since we do not reject any feasible route during the process. In Table 4, there is a comparison between the set partitioning model and the implicit enumeration algorithm plus the revised model. We see that, for the previous five simplified instances, the implicit enumeration algorithm with the revised model provides solutions that are always less expensive, even when the routes obtained travel a slightly longer distance.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Set partitioning model</th>
<th>Implicit enumeration + revised model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Km.</td>
<td>Cost (€)</td>
</tr>
<tr>
<td>210999</td>
<td>2,178</td>
<td>3,664</td>
</tr>
<tr>
<td>220999</td>
<td>2,217</td>
<td>3,589</td>
</tr>
<tr>
<td>100599</td>
<td>1,554</td>
<td>2,680</td>
</tr>
<tr>
<td>121199</td>
<td>2,832</td>
<td>4,163</td>
</tr>
<tr>
<td>301199</td>
<td>2,447</td>
<td>3,982</td>
</tr>
</tbody>
</table>

Table 4: Comparison between the set partitioning model with the implicit enumeration algorithm and the revised model.
The model explained is appropriate for the situation studied, since it produces the relevant outputs at the lowest possible cost and in the time frame required for effective decision making.

A closer examination of the solution method shows that the implicit enumeration algorithm together with the revised model could be used in other environments or problems. The key is to have a distribution problem where the number of truck stops is limited and/or the number of feasible routes remains moderately low, no matter how large the possible route set may be. Problems in which trucks with watertight compartments are involved, or in which goods in bulk are distributed are a clear example (fuel distribution, transportation of milk from farms to processing factories, etc...). In this case, the implicit enumeration algorithm would generate the feasible route set from the possible route set and the revised model would get the optimum route set from the feasible route set.

3 Decision Support System

Once the implicit enumeration algorithm and the model proved to work, we needed to put it all together under some software application, so the logistics manager at the plant could use it.

We developed a software application with Delphi 5.0 using the Dynamic Link Libraries (DLL) distributed with LINGO v6.

We wanted a flexible Decision Support System, so every aspect of the route scheduling process was taken into account, along with the flexibility needed in
a dynamic environment like the one considered.

The solution process with the DSS can be explained as follows:

First, the user can select the set of orders that are to be delivered (Figure 5). At any moment, the user can add or subtract orders from the set; additionally, orders from other days can be selected, which gives an enormous flexibility to the logistics manager.

The user can change at any moment the default distance between each town pair or the possibility of travelling between each pair of towns. By doing this, we can consider many situations such as: asymmetric travelling distances, travelling to a town but not back, bad road conditions, bridges, tunnels and more. (Figures 6 and 7).

After selecting the orders and changing the distance and/or accessibility data (if needed) the user can run the solver, a very useful window shows the progress of the different stages of the solution process (Figure 8).

When the model is solved, a solution screen is shown (Figure 9).

At this stage, the user can select which routes are good and which are not, he might even add or subtract orders from the initial set. At every stage, data can be changed. The DSS responds to data changes, as well as to condition changes with a color coded button at the toolbar showing that the last optimum solution obtained might have changed under the new conditions. With this tool, the user can make what-if analysis and determine which orders are “expensive” to deliver in a given day, or he/she can obtain a more refined solution after a number of tries, or by postponing an order if it is too expensive to serve in a given day, or by serv-
ing future orders in the present day if it is less expensive than to do it separately. Furthermore, the DSS contains all the coding necessary to export the results to common formats, such as Word documents, Excel spreadsheets, RTF text or HTML table pages.

We tested the DSS for 10 different days, which were picked randomly from those of which we had historical data, so we could compare the actual route schedules obtained by the DSS and the routes that were actually scheduled. The resulting graphs can be seen in Figures 10 and 11.

The results show that the company could have saved from 9% to 11% in operational costs and from 7% to 12% in the distance travelled by trucks. This is a worst-case scenario, because the logistics manager could have used the DSS to enhance and refine the solution much better than us, as we only did single run tests, without using all the possibilities of the DSS.

4 Conclusion

In the present paper, we aimed at solving a real problem in a real life company by providing a Decision Support System tool for effective decision making.

The model used by the tool is simple, in the way that the model is easily understood by the decision maker. The model is also complete, easy to handle, and easy to communicate with other software as well as user friendly (Daellenbach, 1994).

We showed that, given the correct generation of the variables of a model (feasible routes), a real life problem can be solved, with only minor simplifications, to an
optimal solution, in computing times under 1 minute in most of the cases.

The logistics manager can now use the DSS and have accuracy and speed over the route scheduling process, allowing him or her to improve the solution and to better control the rush orders by adding them and by re-running the DSS, subsequently obtaining a new optimum solution in a very short amount of time.

The worst case scenario shows that by using the DSS, annual savings of 180,000 € can be obtained only in the company’s plant located in Valencia, Spain. This of course does not include the benefits of the iteration and refinement that the logistics manager can achieve with the DSS or other non-tangible benefits like better customer satisfaction, lower response times to rush orders, availability of accurate information, flexibility and lower manpower requirements.

Furthermore, the company’s plant in Valencia is the largest of the 11 factories of Nanta, so the DSS could be used in every factory, possibly multiplying the estimated savings by a factor of 10.

The proposed two-phase approach could be used in other similar transportation problems, in which trucks with watertight compartments are used. Furthermore, the addition of time windows to the problem is trivial, since we would simply need to have a travel times matrix, -similar to the distance matrix-, and to modify the implicit enumeration algorithm in order to take into account travelling times for satisfying time windows.
Acknowledgments

The authors would like to acknowledge the support and collaboration received from D. Antonio Calatayud and D. Juan LaGuarda, from the Nanta factory located in Valencia, Spain. The authors also thank two anonymous referees and an associate editor for their valuable suggestions to improve this work. We would also like to thank the Foreign Language Co-Ordination Office at the Polytechnic University of Valencia for their valuable help in revising this paper.

References


Figure 1: Two hypothetic routes running very close together in two adjacent zones (zones 3 and 5).
Figure 2: A much better hypothetic solution obtained by coalescing the two previous routes into a single one.
POSSIBLE ROUTES: 15

Figure 3: Possible routes for the example.
FEASIBLE ROUTES: 5

1ST STOP  |  2ND STOP  |  3RD STOP  |  ROUTES GENERATED

A          |           |            | A(100,24)

B          | C          |            | B(120,16), B-C(340,24)

C          | B          |            | C(140,8), C-B(365,24)

Figure 4: Feasible routes for the example.
Figure 5: Order selection screen showing the orders to be arranged into routes.
Figure 6: Distances screen, in which the user can change the distance between the clients.
Figure 7: Accessibility screen, in this screen the user can select which transitions between clients are allowed.
Figure 8: Solution in process screen, which shows the current state of the DSS while solving the problem.
Figure 9: Solution screen; routes obtained by the DSS after the optimization process.
Figure 10: Comparison between the distance in Km. travelled by the routes scheduled and the routes obtained by the DSS.
Figure 11: Comparison between the cost of the routes scheduled by the company and the routes obtained by the DSS.