UNIVERSITY OF MAURITIUS RESEARCH JOURNAL – Volume 16 – 2010
University of Mauritius, Réduit, Mauritius

Research Week 2009/2010

Using Tabu Search Heuristics in Solving the Vehicle Routing Problem with Time Windows: Application to a Mauritian Firm

J. Cheeneebash *
Faculty of Science
University of Mairitius
Reduit
Email: jayrani@uom.ac.mu

Caroline Nadal

Paper Accepted on 03 August 2010

Abstract

In this paper a Tabu Search algorithm has been developed to solve the Vehicle Routing Problem with Time Windows where the capacity of vehicle has been taken as a constraint and the time window within which a customer has to be serviced. The Tabu Search algorithm has been applied to a real life situation for a Mauritian company. The optimal routes of the delivery of frozen chicken were found and it was then possible to minimize the cost of transport within the required the time constraint.

Keywords: Vehicle Routing Problem, Tabu Search method

**For correspondences and reprints*

1. INTRODUCTION

Vehicle Routing Problems (VRP) form part of many day to day activities, from the most complex company such as DHL to a simpler local postal service. Good scheduling is of prior importance to ensure satisfaction of customers. In most organisations nowadays, maximisation of profit remains the main aim. VRP, through good management of vehicles fleet, is a good tool that can allow a business to make savings both in terms of cashflow and time.

Good time management, apart from being cost-effective, is also an important component to ensure quality in customer service. In this paper we take into consideration this temporal aspect of VRP which is referred as the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW consists of finding a set of routes starting and ending at the depot, where is located a homogeneous fleet of vehicles. Adding to these constraints is a set of customers having respective time windows and demands to be satisfied. The primal aim is to minimise the number of vehicles used, and also to reduce the total distance travelled so that the dual of the problem is to maximise profit made.

VRPTW is a hard combinatorial optimisation problem. Combinatorial optimisation problems are concerned with the efficient allocation of limited resources to meet desired objectives when the values of some or all of the variables are restricted to be integral (Hoffman, 1985). Only a few VRPTW have been solved to optimality. Many research papers (Naddef D, and Rinaldi G, 2002; Rousseau LM et al, 2004; Solomon MM, 1987) have been published and most of them have made use of meta-heuristics which enabled them to obtain good results. Among the six known meta-heuristics used to solve VRPTW, Tabu Search has showed the best performance in tackling such problems (Olli Bräysy and Michel Gendreau, 2001).

In this paper we use Tabu Search to solve the VRPTW. Our work is organised as follows: section 2 describes the Vehicle Routing Problem, section 3 explains the Tabu Search. Section 4 gives the methodology used and in section 5 the algorithm is applied to the case of a Mauritian firm. Finally, we conclude about the efficiency of our proposed algorithm in section 6.

2. VEHICLE ROUTING PROBLEMS

VRP can be defined in simple terms, as a problem which consists of determining the optimal set of routes to be taken by a fleet of vehicles in order to route messages, people or goods from one place to another (Toth and Vigo, 2000). VRP are therefore concerned with the distribution management of information, people or goods between depots and customers.

Good management of vehicle fleet for distribution of goods and services is important, as it is estimated that distribution costs account for almost half of the total logistics costs. The distribution costs in some food and drink industries represent 70% of the value added cost of the goods. For instance, both in America and Europe, use of computerised procedures for the distribution process saves the global transportation costs by 5% to 20% (Toth and Vigo, 2000). Thus, use of software to manage vehicle fleet is economically attractive.

In general, the solution of a VRP must be a set of routes used by a single vehicle which starts and ends its journey at the depot, in such a way that demands of customers are satisfied as well as operational constraints. An example of an operational constraint could be that the sum of demands of customers on a particular route should not exceed the capacity of the vehicle using that route. All these operations should be performed in such a way so as to minimise total travel costs (Toth and Vigo, 2000).

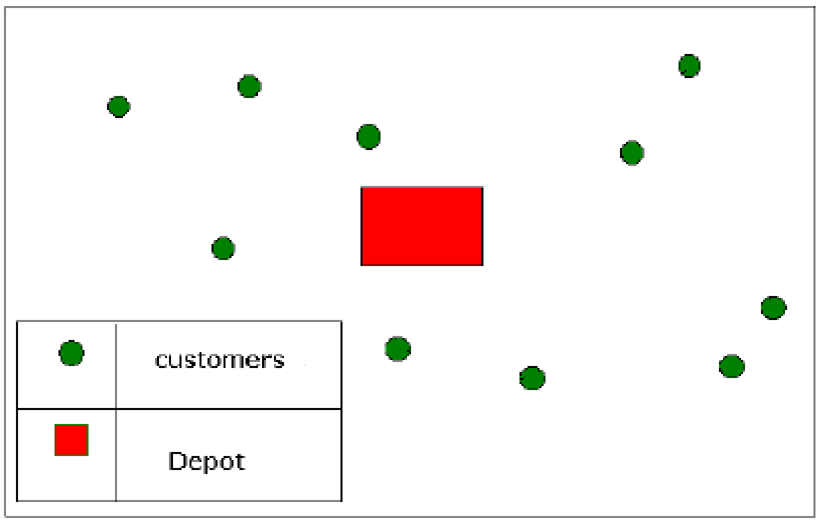

Figure 1: Scattered Customers in a certain area in the depot.

Routes are constructed in such a way so that demands of customers are satisfied as well as the operational constraints, while minimising total travelling costs.

2.1 Problem Definition and formulation

Let $G = (N, A)$ be a complete graph where: $C = \{2, \ldots, n\}$ represents the set of customers located in n different locations and $N = \{1,2, \ldots n\}$ is the vertex set where the vertices $i = 2,\ldots,n$ corresponds to the customers and the vertex 1 represents the depot. A is the arc set where, for each arc $(i, j) \in A$ is associated a non-negative cost dij which represents the cost of travelling from node i to node j.

Toth and Vigo (2000) highlight that usually the use of loop arcs *(i, i)* is not allowed and thus d_{ii} is assigned the value of $+\infty$.

Now, if G is a directed graph, the cost matrix d is asymmetric and the corresponding problem is an asymmetric capacitated vehicle routing problem. If, on the other hand, G is a graph where cost of travelling from i to j, (dij), and vice-versa (dji), are equal for all $(i,j) \in A$, the problem is known as a symmetric capacitated vehicle routing problem (Toth and Vigo, 2000). In this paper, symmetric capacitated vehicle routing problem will be considered.

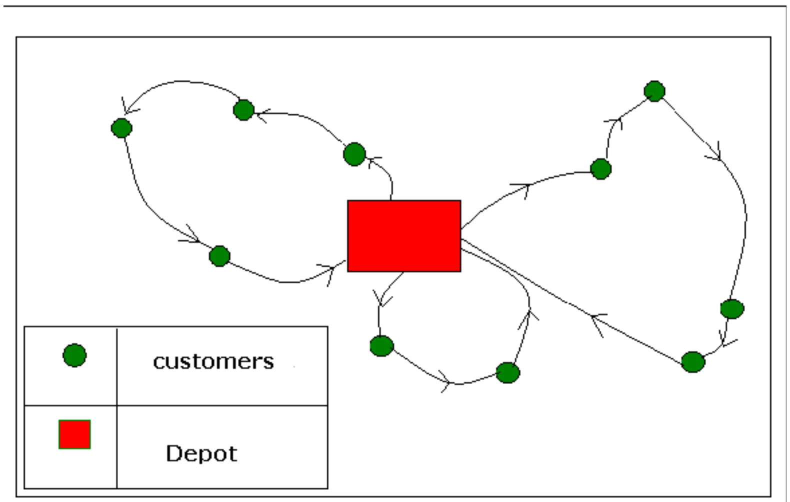

Figure 2: Optimal set of routes joining customers to the depot.

The cost matrix satisfies the triangular inequality:

$$dik + dkj \geq dij \text{ , for all } i, j \text{ , } K \in V.$$

Euclidean distance between any two customers i and j, refers to those vertices associated to points of the plane having given coordinates and cost dij. In this case the cost matrix is symmetric and satisfies the triangular inequality. The problem which is termed Euclidean Capacitated Vehicle Routing Problem will be dealt with in this paper.

The VRPTW is an extension of the capacitated VRP with the added constraint that each customer, say i, must be served within a particular time window often denoted by $[ai , bi]$ where a_i is the earliest time to service customer i and bi, the latest time at which serving customer i is possible.

Also the depot has an assigned time window denoted by *[a1, b1]* which corresponds to the earliest and latest time to leave and return to the depot after servicing customers. It is important to note that time windows can be either soft or hard.

A soft time window allows customer i to be serviced even at a time not found within the interval *[ai, bi]*, at a certain cost. On the other hand, for a hard time window, if the vehicle arrives at a customer i before a_i, it must wait until *ai* to service customer i and similarly the vehicle cannot service the customer i after the latest time *bi*.

Associated to every pair of locations *(i, j)*, apart from the travelling cost *dij* where $i,j \in$ N and $i \neq j$, is the travelling time from i to j denoted by *tij*. Also each customer has a demand denoted by *wi* where $w_i > 0$. It is to be noted that the demand for the depot is zero. We are also provided with a set of vehicles V of identical capacity m.

So our objective is to minimise the total travel cost while satisfying the various operational constraints namely those concerning:

1. The customer, in terms of his demand and time within which he wants to be served.
2. The depot, in terms of earliest and latest time to leave and return back.
3. The vehicle, where the demand of customer(s) on a route should not exceed its capacity.

Table 1. Summary of the variables used for the mathematical formulation

Variables	**Definition**
C	Set of Customers
N	Set including Customers and Depot
d_{ij}	Travelling cost on moving from customer i to customer j, where $i, j \in N$.
$[a_i, b_i]$	Time window of customer i, where $i \in C$.
$[a_1, b_1]$	Time window of Depot

V	Set of vehicles available at depot.
m	Capacity of vehicle.
w_i	Demand of customer i, where $i \in N$.

A mathematical formulation of the above problem will now be presented which has been adapted from Sin C Ho and Dag Haugland (2001). Our problem involves two types of decision variables. A decision variable is an unknown quantity representing a decision that needs to be made. It is the quantity the model needs to determine. The two decision variables are:

1. A flow variable, x_{ijk}, where for each arc $(i\ ,\ j)$, where $i,j \in$ N and $i \neq j$, and for each $K \in$V; we define

$$x_{ijk} = \begin{cases} 1;\text{ if vehicle K travels directly from customer i to customer j} \\ 0;\text{else} \end{cases}$$

2. A time variable, s_{ik} which denotes time at which vehicle K starts serving customer i.

Objective function:

$$\text{Minimise } z(x) = D\sum\sum\sum x_{ljk} + \sum\sum\sum d_{ij} x_{ijk} \qquad (1)$$

Subject to the constraints

$$\sum_{j \in N} x_{ijk} = 1 \quad \forall k \in V$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall\, h \in C, \forall\ k \in V$$

$$\sum_{i \in| N} w_i \sum x_{ijk} \leq m \quad \forall\ k \in| V$$

$$s_{ik} + t_{ij} - k_{ij}(1 - x_{ijk}) \leq s_{jk} \quad \forall\ j \in C, \forall\ i \in N, \forall\ k \in V$$

$$a_i \leq s_{ik} \leq b_i \ \forall\ i \in N, \forall\ k \in V$$

$$s_{ik} + t_{jl} - k_{ij}(1 - x_{ijk}) \leq b_l \ \forall\ i \in C, i \in N, \forall\ k \in V$$

$$s = a_l \ \forall\ k \in V$$

$$x_{ijk} = 0 \forall\ i \in C, \forall\ k \in V$$

$$x_{ijk} \in \{0.1\}\ \forall i, j \in N, \forall\ k \in V$$

3. TABU SEARCH

Tabu Search was first proposed in 1986 by Fred Glover in one of his published papers. Tabu Search was invented because a wide range of hard combinational optimisation problems in fields such as economics, engineering, business and science needed to be tackled within a practical time bound. Tabu Search, according to Glover, is a meta-heuristic that guides a local heuristic to explore the solution space beyond local optimality.

Tabu Search starts just as an ordinary local search, proceeding iteratively from one solution to the next until some stopping criteria is satisfied while making use of some strategies to avoid getting trapped in a local optima. In fact, local search such as the descent method for example, often results in a local optimum which may or may not be a global optimum, which is one that minimises *f(x)*, the objective function, for all $x \in \boldsymbol{X}$**.**

To each $x \in \boldsymbol{X}$**,** where $\boldsymbol{X}$ represents the constraints on the vector of decision variable *x*, is associated a neighbourhood $\boldsymbol{N(x)} \subset \boldsymbol{X}$**,** where each solution $x'' \in \boldsymbol{N(x)}$ is reached from *x* by a move operation (Glover and Laguna, 1997).

The characteristic feature of Tabu Search that distinguishes it from other search methods is its use of memory which can be qualified as both explicit and attributive. Explicit memory stores complete solutions and is composed of elite solutions visited during the search. Interesting unexplored neighbours of elite solutions can as well be recorded by making use of an extension of this memory. Local search is extended through use of memorised elite solutions or their attractive neighbours (Glover and Laguna, 1997). Attributive memory, conversely, is used for guiding the search process. Information about changing solution attributes arising on moving from one solution to another, are stored. An example of attribute can be the nodes and arcs that are deleted, removed or repositioned in a graph by the moving mechanism.

3.1 Features of Tabu Search algorithm

The primary parameters required for running the Tabu Search algorithm are summarised in figure 3.

3.1.1 Search space

It is simply the space containing all feasible solutions that are relevant to our problem and that will be taken into consideration during the search. In our problem, the search space comprises of all feasible routes that satisfies all the given constraints stated previously in our formulation.

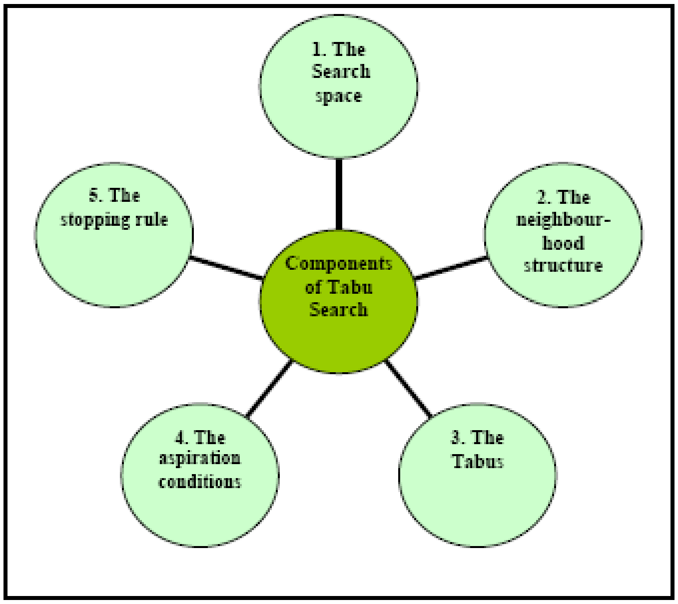

Figure 3: Components of the Tabu Search.

3.1.2 The Neighbourhood structure

The Neighbourhood structure is created by applying a "move" to the current solution *xt* at each iteration thereby creating a neighbourhood *N (xt)* in the search space. There exist, according to Brasssy et al (2001), two types of neighbourhood structures namely:

1. Simple Neighbourhood structure
2. Complex Neighbourhood structure

The simple neighbourhood structure is created by making simple "move" operations. For instance, it involves moving only one customer at each iteration from one route and placing the latter in another route or somewhere else in the same route. For example, in the diagram below, customer 2 has been moved from its current position (between customer 1 and customer 3) and has been inserted between customer 3 and 4.

On the other hand, constructing complex neighbourhood structures involve more complicated move operations. A few examples of such moves include relocate operator, exchange operator, 2-opt* operator or even a combination of them. In the next diagram, exchange move operator has been illustrated. In this example customer 2 has been moved from route 1 and transferred to route 2 and at the same time customer 4 has been moved to route 1.

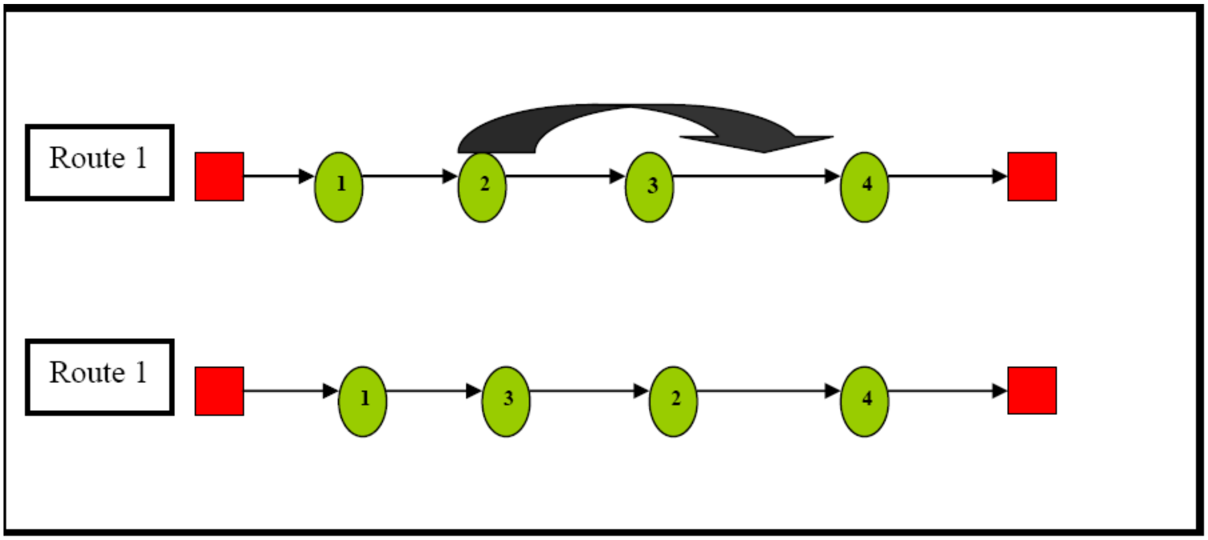

Figure 4(a): Example of a simple move.

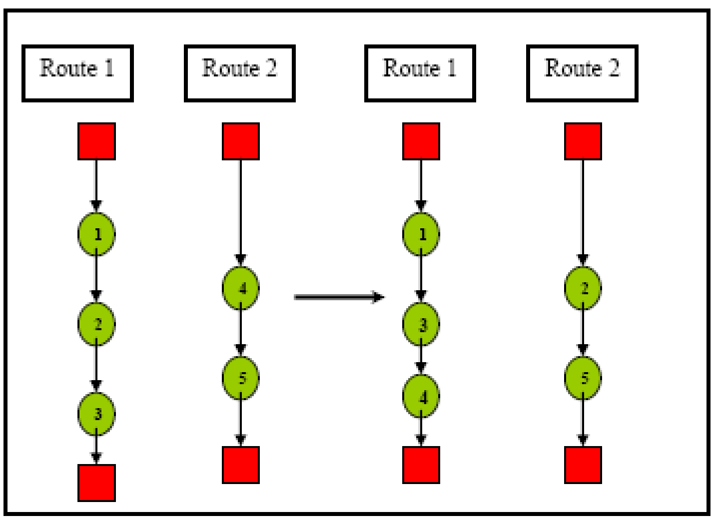

Figure 4(b): Example of an exchange move operator.

3.1.3 Tabus

"Tabu" is the fundamental feature of Tabu Search which distinguishes it from Local Search. In fact "Tabus" are used in order to prevent cycling when moving away from local optima through non-improving moves (Brasssy et al 2001).

For example, if customer 2 has been moved from route 1 and placed in route 2, moving customer 2 back to Route 1 can be declared Tabu for a number of iterations.

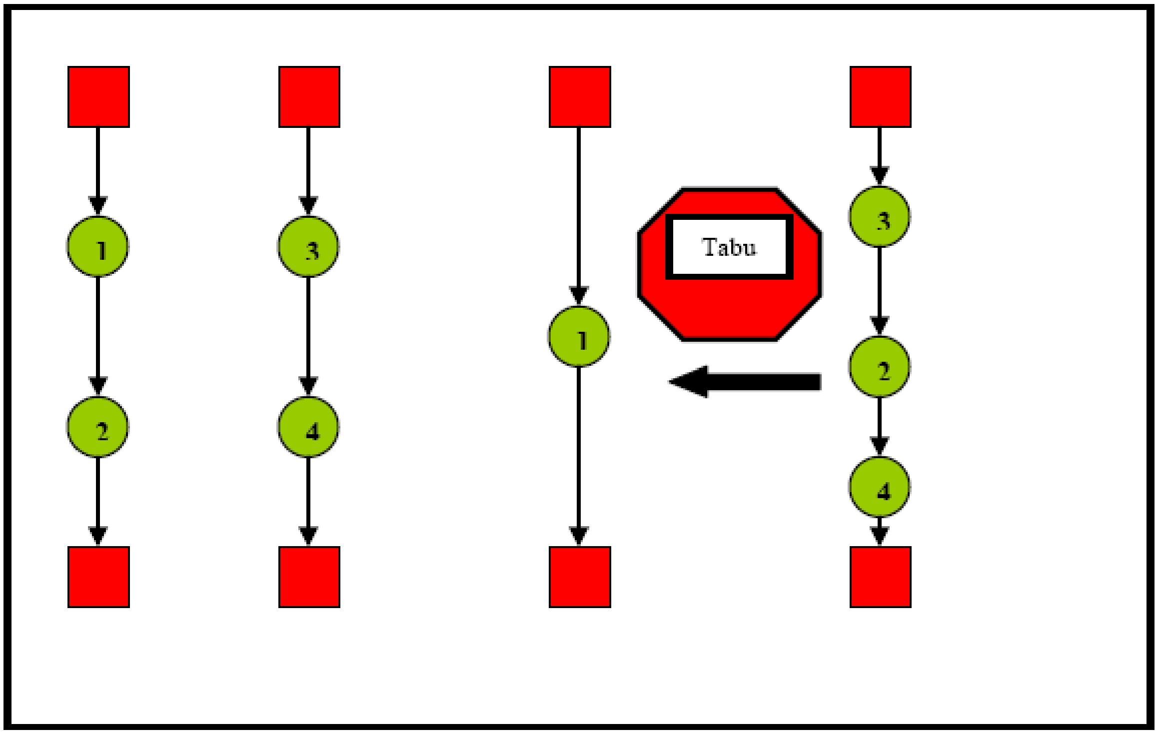

Figure 5: Example of a Tabu move.

Tabu tenure refers in fact to the amount of iterations for which a move is declared Tabu. Tabus are stored in short-term memory structures known as tabu lists. This storage can be done in a number of ways namely:

1. Recording complete solution
2. Recording the last few transformations performed on the current solution and preventing reverse transformations.

Tabus do have some drawbacks though. It can prohibit interesting moves from occurring though that move may have no danger of cycling back. It may as well lead to an overall stagnation of the searching process.

3.1.4 Aspiration Criteria

According to Brasssy et al (2001), an aspiration criterion is an algorithmic device that will enable one to cancel Tabus. The simplest aspiration criteria involve allowing a move, even though being Tabus, if it results in a solution with a larger objective value than the current best-known solution.

3.1.5 Stopping conditions

At some stage the iterative search process must terminate. In a Tabu Search algorithm, there are a number of ways of ending the search process such as:

1. After a fixed number of iterations.
2. After some number of iterations without an improvement in the objective function value.
3. When the objective reaches a pre-specified threshold value.

3.2 Strategies used in Tabu Search

In order to refine the search procedure, intensification and diversification strategies are used.

3.2.1 Intensification

Intensification strategies, according to Laguna et al (1999), are based on modifying choice rules to encourage move combinations and solution features historically found good. Closely linked to the implementation of intensification strategies is explicit memory; this is because, to examine neighbours of elite solutions, the latter must be recorded. In order to allow intensification process to occur, an additional term can be added to the objective function that will restrict the search to solutions close to the current one. It is advisable that intensification be performed during a few iterations and afterwards another region of the search space is explored.

3.2.2 Diversification

Diversification, in contrast, extends the search procedure to unvisited portions thereby generating new solutions not encountered beforehand. In order to induce the diversification process, a supplementary term can be added to the objective function which will penalize elements close to the present one.
The modified objective function, *O* becomes:

$$\bar{O} = O + \text{Diversification} + \text{Intensification}$$

3.3 General formulation of Tabu Search algorithm

Below is a general formulation of the Tabu Search algorithm (Li & al., 2005).

Step1: Choose an initial solution *i* in *S*.

Set i* = *i* and *k* = 0.

Step 2: Set *k = k + 1* and generate a subset *V** of solution in *N (i, k)* such that either one of the tabu conditions is violated or at least one of the aspiration conditions is satisfied.

Step 3: Choose a best *j* in *V** and set *i= j*.

Step 4: If *f (i) < f (i*)* then set *i* = i*.

Step 5: Update tabu and aspiration conditions.

Step 6: If a stopping condition is met then stop. Else go to step 2.

Notations:

i, j: solution indexes.

k: iteration index.

*V**: subset of solution.

N (i, k): neighbourhood of solution i at iteration k.

f(i): Objective function value for solution i.

4. METHODOLOGY

A two-phase approach in order to solve the problem at hand (Toth and Vigo, 2000). A two-phase heuristic involves creating in a first stage an initial feasible solution using a construction heuristics and in a later stage, an improvement heuristic is applied to that initial feasible solution obtained previously in order to find an optimal solution to the given VRPTW.

The construction stage comprises of inserting into a set of feasible routes, all the customers to be serviced, while ensuring that minimum distance is covered. The improvement heuristic, on the other hand, entails, moving at each iteration from that set of feasible routes to its feasible neighbourhood until some stopping criteria is satisfied.

For the given problem being tackled, an initial solution was obtained according to an algorithm developed by Sin.C.Ho and Dag Haugland (2001) which is based on a simple analysis of travelling time and waiting time. For the purpose of this project, the algorithm was adapted to suit the constraints of the latter.

4.1 Creation of an Initial Solution

The initial solution is determined by finding the nearest unrouted customer, $\hat{j}$ to i, the latest routed customer, where the criteria to be satisfied for choosing $\hat{j}$ is that; $\hat{j}$ minimizes the sum of travel time and waiting time from i to $\hat{j}$ as per the equation below:

$$\hat{j} \in \arg\min_{j \in C''} \{t_{ij} + \{a_j - \theta_i - t_{ij}, 0\}\}$$

where θ_i is the time at which customer i is serviced and C' represents the set of unrouted customers.

This process is recurred for a number of iterations until all the customers are inserted into an appropriate route. The algorithm below gives a more detailed explanation of how this initial solution is generated.

4.2 Improvement phase using Tabu Search

Before going more in depth into the next step that is, the improvement phase, details concerning features of Tabu Search used for the improvement step of the problem being dealt with will be considered.

4.2.1 Neighbourhoods

Neighbourhoods were created by swapping a pair of customers within the same route or in between routes by using three types of move operators namely: 2-opt* operator, exchange operator, 2-opt operator which are shortly described below. For each of these move performed, the move value was calculated, where the move value refers to the difference in the objective function value before and after the move.

Algorithm 1

K = 1
Repeat

Start with an empty route K starting from the depot

Initially, $\hat{j}$ = 1(as each route must start and end at the depot) and $\theta_1 = a_{11}$, where a is the column matrix representing earliest time to service each customer.

Repeat

Set $i = \hat{j}$

Find the nearest unrouted customer, $\hat{j}$ to i where the criteria to be satisfied for choosing that unrouted customer are that, its demand is not equal to zero and that the unused capacity of that vehicle exceeds the next customer's demand. At the same time, $\hat{J}$ must satisfy equation 1 above and be feasible in terms of time constraints.

$\hat{J}$ is then inserted into that route K and the new value of $\theta_{\hat{j}}$ is given by:

$$\theta_{\hat{j}} = \theta_i + t_{ij} + \max\left\{a_{\hat{j}} - \theta_i - t_{i\hat{j}}, 0\right\}$$

The capacity left after insertion of customer, $\hat{j}$ is then computed.

Until *Capacity left = 0 or no more insertions are valid.*

K = K + 1

Until *all customers are routed.*

The 2-opt* operator has been used to swap a pair of customers between two routes. This move has been illustrated below. In the diagram below, customer 3 has been moved from its initial position (between customer 4 and the depot) and placed between customer 1 and the depot on another route. Similarly for customer 2, who has been removed from his original route and placed in another route between customer 4 and depot.

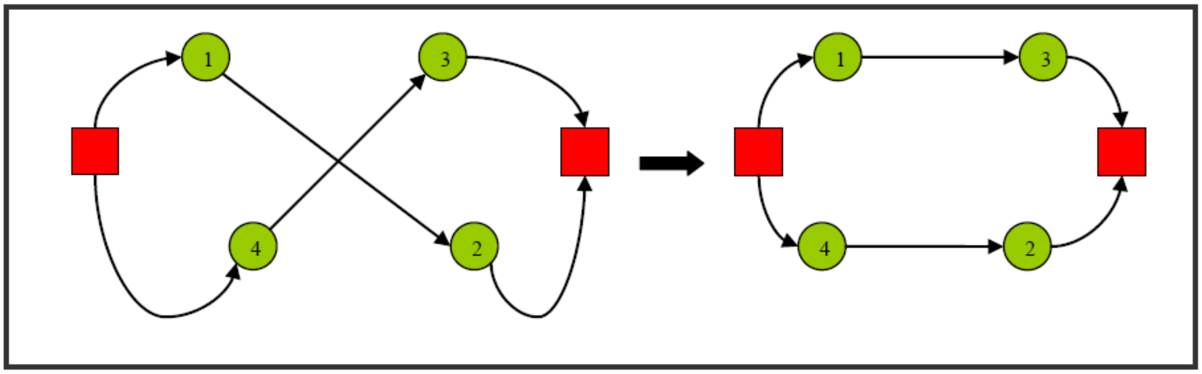

Figure 6: 2-opt operator.*

The exchange operator has also been used to generate neighbourhoods. Just as, the above move, this move allowed swapping of customers between routes. In the diagram below, customers 6 and 5 have been swapped from their initial routes.

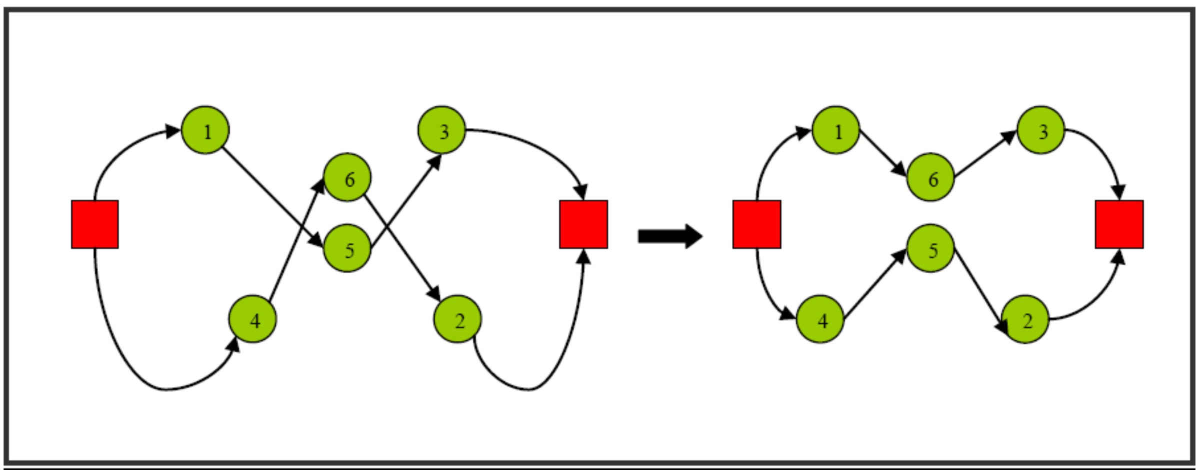

Figure 7: Exchange operator.

The 2-opt move involved swapping a pair of customers found within the same route. This is illustrated below. Instead of servicing customers 1, 2, 3 then 4 after leaving depot, by swapping customers 2 & 3, the new order in which the customers are serviced is given by: 1, 3, 2 then 4.

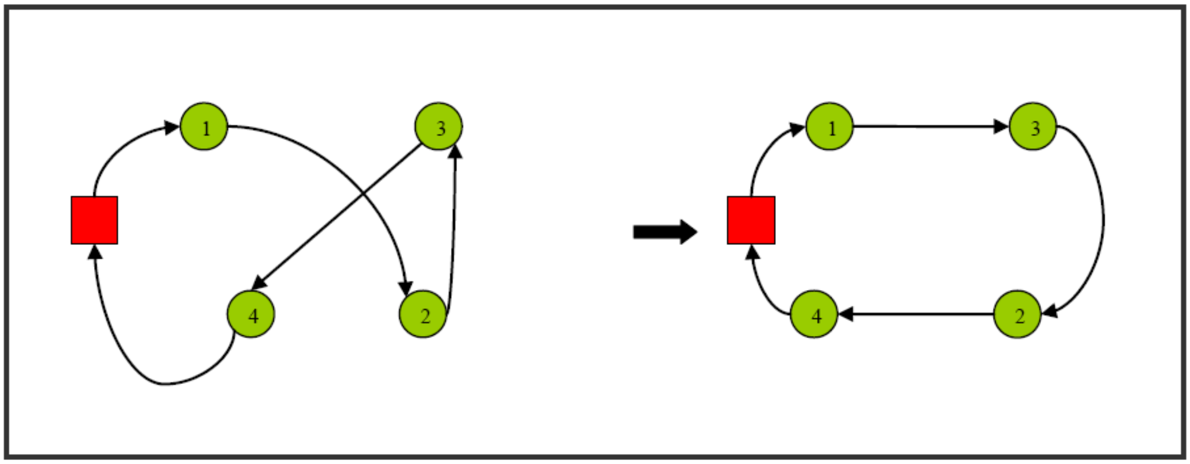

Figure 8: Exchange operator.

For each move described above, the move value is calculated. The move value has been defined as the difference between the objective function value of the initial solution obtained in algorithm 1 and the objective function value of the new neighbourhood generated through each move.

Another important feature to be described is the acceptance criterion.
The acceptance criteria defined for the Tabu Search algorithm are that each of the neighbourhoods created for a given solution, must obey the given constraints that are:

1.) Capacity on a given route should not exceed vehicle capacity and,
2.) Feasibility in terms of time constraints.

For the VRPTW, there exist two types of acceptance strategies namely; the first accept strategy and the best accept strategy. The first accept strategy scans the neighbourhoods of a given solution in search of the first neighbour satisfying the predefined acceptance criteria. The best acceptance criterion, on the other hand, selects the move with the highest move value among all the moves in the neighbourhood. For the purpose of this paper, first acceptance criteria have been used in the algorithm.

4.2.2 Stopping Criteria

The stopping criterion used to terminate the Tabu Search process is preset after a fix number of iterations.
Algorithm 2 below gives a better idea of how the Tabu Search procedure was used to solve that VRPTW to optimality. It was based on a paper published by Laguna et al. (1998) and adapted to the context of our problem. The choice of the variable *MAXINT* and *MAXDIV* was chosen using the same criteria mentioned in the paper Laguna et al. (1999).

5. Application to a Mauritian Context with Panagora Marketing Co Ltd

For the purpose of this paper, focus will be made solely on the branch of Panagora Marketing Co. Ltd distribution activities that is concerned with the selling and distribution of frozen chicken, produced from the chicken farms of the FAIL group. The firm has around 450 clients spread throughout the island. The system of distribution used is zone-wise and based on **time window delivery**.

For serving these customers, there are some restrictions imposed on the delivery vehicle in order to satisfy the requirements imposed by the Hazard Analysis Critical Control Path (HACCP) food hygiene standard. Firstly the vehicles have to be equipped with a refrigerated cooling system. The acceptable temperature range is from 0ºC to 6ºC, beyond which products cannot be delivered to clients. In order to ensure that products are maintained at the required temperature, computerized temperature controls are used. Another restriction is that the vehicles can only be filled up to a maximum of 70-75% capacity.

Algorithm 2

An initial solution is obtained from ***algorithm 1.***

Initiate all parameters

Moves are generated according to the moves defined above.

Feasible neighbourhoods are stored by storing their corresponding move values.

Intensification process starts

Initially move value = 0

For *MAXINT iterations*

Choose a customer at random; where probability of choosing a customer is proportional to its demand.

Move with highest move value for that customer is selected.

For *Tabu tenure iterations*

Customer is placed in tabulists and thus cannot be chosen

End

Initial move value = move value

End

The number of times each customer has occurred in tabulist is calculated and is referred to as the frequency of that customer.

Diversification process starts

For *MAXDIV iterations*

Customer is randomly selected where probability of selecting a customer is inversely proportional to the frequency of that customer.

Chosen customer is placed in best position having highest move value

End

5.2 Adapting proposed algorithm to the actual problem.
The problem consists in that case, of finding the minimum set of routes such that, respective customer's demand is satisfied within a time frame which has previously been agreed on a contract by the clients and the company.

Data has been collected for one week and the algorithm proposed was applied to each daily routing in order to find the possible savings that could be made by the company. The possible profits made over one month, assuming the company does not acquire any new client during that period, was then calculated by extending the profit made over a week to one month.

There were between 60 and 70 customers to route each day. These customers were grouped in regions according to their geographical positions and as such, the algorithm consisted of finding the optimal route set from the depot to these regions and back to the depot.

As such, Global feed distance calculator, a software available online, was used as a tool to calculate the road distances between each and every region as well as their distances from the depot and each value obtained

For the proposed algorithm, it has been assumed that the distance between any two customers i and j, (d_{ij}) where i and j are found in the set N equals to the distance between j and i, (d_{ji}) denoted by:

$$d_{ij} = d_{ji}, \quad \text{for all } i, j \text{ found in } N$$

This might not occur in real life as cases might arise that roads are one-way, for example, on the motorway (Toth & Vigo, 2000). Also, it has been assumed that the traffic is fluid, in other words, no traffic jam occurs on the road.

For each region, the earliest time was calculated by considering the earliest time the first customer of that region needed to be serviced while the latest time to leave that region was obtained by considering the latest time that the last customer in that region needed service.

Also, the total demand for a particular region was calculated by calculating the individual demand of each and every customer in that region.

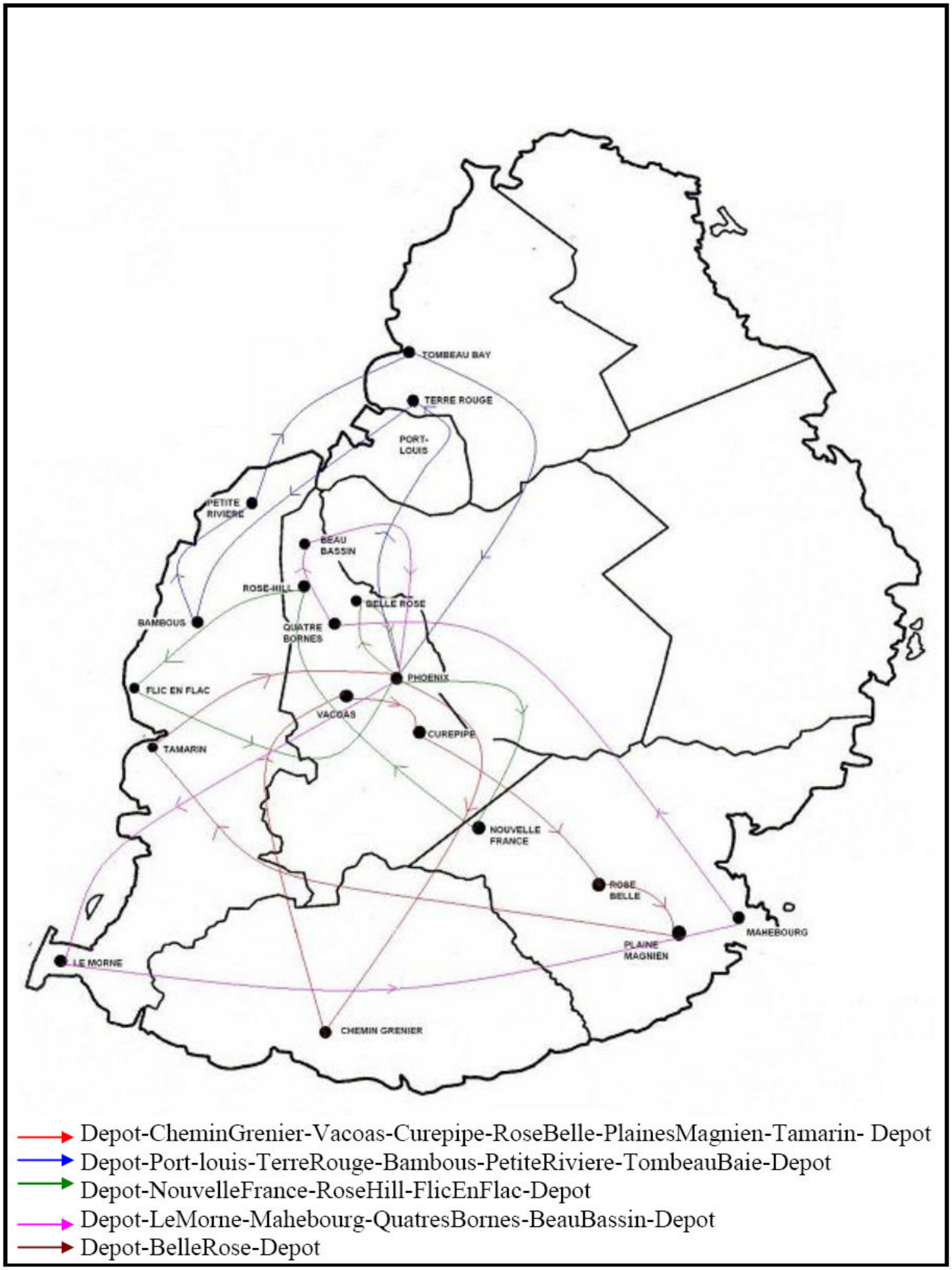

Figure 9. Map showing the different routes constructed about the depot, Phoenix.

The capacity of the vehicle also had to be calculated as the vehicle can hold only 75% of its capacity due to certain norms of hygiene as stated earlier. In addition, it has been assumed that the fleet was a homogeneous one though slight differences exist in their capacities.

The algorithm was then run using Matlab 7.0 and for each day, an optimal set of routes was obtained. Unfortunately in this paper we did take into account the computational cost. Each constructed route satisfied time window constraints and comprised of regions whose total demand did not exceed the vehicle's capacity. The map on the next page shows the different routes obtained for a particular day. The depot is located at Phoenix and each route starts and ends in that region.

The number of vehicles used for each day, the percentage time window satisfied as well as the distance covered using the proposed algorithm and their corresponding values in real life were tabulated as shown below. This has been done in order to evaluate the amount of profit that could be made over a week if the company had used a scheduling similar to the solution of our algorithm.

Table 5.3: Feasible profit over a one week period.

Table 5.3: Feasible profit over a one week period

Day	Number of vehicles used (In real life)	Number of vehicles used (Using my program)	Percentage of time window satisfied (In real life)	Percentage of time window satisfied (Using my program)	Distance covered (In real life)	Distance covered (Using my program)
Monday	8	5	95%	95%	519.96	517.2300
Tuesday	10	6	95%	95%	595.78	604.3800
Wednesday	9	5	95%	95%	641.99	497.9600
Thursday	9	5	95%	95%	479.47	409.5200
Friday	9	7	95%	95%	815.24	555.3600

As can be seen from the table, the company can reduce the amount of distance travelled from 3,052.44 km to 2,584.45 km for one week. And for one month, the amount of distance travelled that can be saved is around 1,871.96 km. This decrease in the amount of distance travelled reduces the amount of fuel used by the vehicle and thus benefits the company. Also, travelling less causes less wear to the vehicle being used which is again profitable to the company. In addition, driving less can prove to be more ecological as it reduces the amount of exhaust gases emitted.

Secondly, the amount of vehicle used for each day has also decreased indicating that labour can be saved. At Panagora, the drivers do not have their lunch time as such. They have as priority to complete their entire route and return to the depot. In any circumstance, whether they perform overtime or not, they will be paid two hours overtime daily as per their contract with the company. But as can be seen in the illustration on the map, there will be uneven distribution of tasks performed by each driver. For instance, one driver might find himself performing a distance of 141.91 km in a journey, as indicated by the red arrows on the map, while another one might have to perform a distance of 11.64 km, as indicated by the brown arrow, in a journey. This algorithm did not take into consideration driver's constraints explaining the occurrence of these uneven routes.

Concerning the time window, it can be noted that only 95% of the customers were serviced within their time frames. The remaining 5% represents the depot. This is due to the fact that some vehicles returned to the depot after b1.

The slight decrease in travelling cost indicates the importance of scheduling. Good scheduling can be beneficial to the company in a number of ways; it saves time and unnecessary labour, and it can prove to be economical as well as ecological.

It is good to note that the day to day distribution at Panagora Marketing Co. Ltd is quite a good route planning though. This is because the distances that could be saved, are not that huge and the proposed algorithm includes many assumptions that may not hold in real life situations.

It is important to note that the proposed algorithm could be improved in a number of ways. For instance, it has been assumed that the company does not acquire any new customers in the incoming weeks but this might not occur in real life. So, instead of considering a static VRPTW as has been done, a dynamic VRPTW that allows for new customers could have been implemented.

6. CONCLUSION

A Tabu Search algorithm for solving VRPTW has been developed in two steps. As a first step, an initial feasible solution satisfying both capacity and time window constraint has been generated. To improve that initial set of routes, Tabu Search was applied. This algorithmic process fine-tuned the set of routes acquired in order to obtain an optimal set of routes. Some future work that can be done is to test the performance of the algorithm by varying the number of clients and the length of the routes.

The algorithm which was run on Matlab 7.0 to applied to a sample of 25 customers of the Solomon benchmark problem. The results showed that the optimal solution was less than the initial solution thereby showing that the distance travelled had decreased and thus, the profit made increased by a certain amount.

The same two-phase algorithm was adapted to a real-life situation, and this time, it was observed that the distance that could be covered using that developed algorithm decreased compared to the amount of kilometres travelled in reality. From that real-life application of the proposed algorithm, it was also noted that good scheduling of vehicles could prove to be beneficial to the company in terms of pollution prevention as well as time and financial savings, especially in the economic crisis in which the world is going through nowadays. For Panagora Marketing Co. Ltd, such savings could compensate a portion of the amount of diesel used to run the vehicle's refrigerator for example, or to compensate for the capacity of the vehicle lost due to hygiene norm restrictions.

7. REFERENCES

Braysy, O & Gendreau M, 2001, 'Tabu Search heuristics for the Vehicle Routing Problem with time windows', *www.neo.lcc.uma.es/radi-aeb/WebVRP/data/articles/Metahur VRPTW.doc* [Accessed on 10th January 2009].

Glover, F & Laguna, M,1995, 'Tabu Search, chapter 3', *In: Reeves, C.R.(ed) Modern heuristic techniques for combinatorial problems, McGraw-Hill* pp. 70-150.

Hoffman, K, 1985 'Combinatorial and Integer Optimisation', www.iris.gmu.edu/~khoffman/papers/newcomb1.html. [Accessed on 8 April 2009].

Laguna, M, Marti, R & Campos, V, 1999,'Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem', Computers & Operations Research, 26 , 1217-1230.

Li, L, Liu, H & Lu, R, 2005, 'Tabu Search, www.courses. Washington.edu/inde510/Tabu Search, Accessed on 25 November 2008.

Naddef, D., and Rinaldi, G. 2002. 'Branch and cut algorithms for the capacitated VRP', The Vehicle Routing Problem (P.Toth and D. Vigo, eds), SIAM Monographs on Discrete Mathematics and Applications, 53-84, SIAM, Phildelphia.

Rousseau, LM., Gendreau, M., and Pesant, G. 2004, 'Solving VRPTWs with constraint programming based column generation', Annals of Operations Research, 130, 199-216.

Sin, C, Ho & Haugland, D, 2001, 'A Tabu Search heuristics for the Vehicle Routing Problem with time windows and split deliveries', www.neo.lcc.uma.es/radi-aeb/WebVRP/data/articles/Metahur VRPTW.doc . [Acessed 10 September 2008].

Solomon, MM. 1987, 'Algorithms for the vehicle routing and scheduling problems with time window constraints', Operations Research, 35, 254-265.

Toth, P., Vigo. D, I 2000, 'An overview of Vehicle Routing Problems', *Vehicle Routing Problems*, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, chapter