Improving on the initial solution heuristic for the Vehicle Routing Problem with multiple constraints

J. W. Joubert
Department of Industrial and Systems Engineering, University of Pretoria

Abstract

The integration of multiple constraints of the *Vehicle Routing Problem* (VRP) variants is computationally expensive. Although vehicle routing problems have been well researched, variants are typically treated in isolation, whereas industry requires integrated solutions. The concept of time window compatibility is introduced to improve the initial solution algorithm when combining variants such as a heterogeneous fleet, double scheduling, and multiple time windows. The paper reports on the algorithm's results when benchmark test data is used. The results relate to both the computational burden of the algorithm, as well as the quality of the initial solution.
Keywords: vehicle routing, VRP, initial solution, heuristics.

1 Introduction

Vehicle routing and scheduling problems are well-researched in the field of *Operations Research*. The main objective of these types of problems are to minimize the distribution costs for individual carriers. Given the complexity of the type of problem, extensive research has been conducted to develop exact and heuristic solution techniques for urban distribution problems.

The *Vehicle Routing Problem* (VRP) can be described as the problem of assigning optimal delivery or collection routes from a depot to a number of geographically distributed customers, subject to side constraints. In its basic form, the VRP can be defined with $G = (V, E)$ being a directed graph where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of vertices representing customers, and with v_0 representing the depot where m identical vehicles, each with capacity Q, are located [14].

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

$E = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ is the edge set connecting the vertices. Each vertex i, except for the depot $(V \backslash \{v_0\})$, has a non-negative demand q_i and a non-negative service time s_i. A distance matrix $C = \{c_{ij}\}$ is defined on E. In some contexts, c_{ij} can be interpreted as travel cost or travel distance. Hence, the terms distance, travel cost, and travel time are used interchangeably. The VRP consists of designing a set of m vehicle routes having a minimum total length such that

- each route starts and ends at the depot,
- each remaining vertex $(V \backslash \{v_0\})$ is visited exactly once by one vehicle,
- the total demand of a route does not exceed Q, and
- the total duration (including service and travel time) of a route does not exceed a preset limit L

The VRP is a hard combinatorial (np-hard) optimization problem for which Laporte [8] has indicated several exact and approximate solution algorithms. An np-hard problem implies that the solution space will increase at an exponential or factorial rate (non-polynomial) as the number of customers/vertices increases. Early researchers such as Clarke & Wright [2] realized that exact algorithms can only solve relatively small problems, but a number of heuristic (near-optimal) algorithms have proved very satisfactory.

1.1 Solution algorithms

Heuristics typically uses a greedy approach to obtain a good initial solution through a tour-building algorithm in efficient time, and then incrementally improve the solution by neighborhood exchanges or local searches.

Solomon divides VRP tour-building algorithms into either sequential or parallel methods [11]. Sequential procedures construct one route at a time until all customers are scheduled. Parallel procedures are characterized by the simultaneous construction of routes, while the number of parallel routes can either be limited to a predetermined number, or formed freely. Solomon concludes that, from the five initial solution heuristics evaluated, the *sequential insertion heuristic* (SIH) proved to be very successful, both in terms of the quality of the solution, as well as the computational time required to find the solution.

Improvement heuristics tend to get trapped in a local optimal solution and fail to find a global optimum. Heuristics have evolved into *global optimization heuristics*. These are general master strategies to solve problems, and are based on intelligent search techniques. Where heuristic searches are limited to steps that will improve the objective function, global optimization heuristics allow steps that will temporarily decrease the objective function value, in an attempt to escape the local optimum and look for the global optimum, or at least a better local optimum. These global optimization heuristics are often called *metaheuristics* because the procedure used to generate new solution out of the current one, is embedded in a heuristic which determines the search strategy. The main drawback of metaheuristics is that they do not have definitive stopping criteria; the longer the computation time, the higher the probability of finding the global optimum [14].

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

Tan *et al.* [13] compare the three popular metaheuristics, namely *Simulated Annealing*, *Genetic Algorithm*, and the *Tabu Search* methods with one another and concludes that there is no single heuristic that is generic enough to solve problems for all situations. Instead, they are inevitably problem specific.

The focus of this paper is to find an improved initial solution, as Van Breedam [14] notes that the performance of metaheuristics such as the Tabu Search is highly dependent on the quality of the initial solution.

1.2 The underlying assumptions of the *VRP*

The basic VRP makes a number of assumptions, including utilizing a homogeneous fleet, a single depot, and one route per vehicle. These assumptions can be eliminated by introducing additional constraints to the problem, and hence a specific *variant* of the VRP problem is created. This implies increasing the complexity of the problem, and, by restriction, classifies the extended problem variant as an np-hard problem. It is the believe of the author that the various VRP variants have been researched in isolation, few examples exist where a reasonable number of applicable variants have been integrated into a single algorithm. By *applicable* is implied appropriate to the South African routing environment which have a distinct particularity. Joubert [6] elaborates on the implications of integrating multiple time windows, a heterogeneous fleet, and double scheduling into a single initial solution algorithm.

2 Time window compatibility

Under Solomon's [11] sequential insertion heuristic, *initialization criteria* refers to the process of finding the first customer to insert into a route. The most commonly used initialization criteria is the *farthest unrouted* customer, and the customer with the *earliest deadline*, or the earliest *latest allowed arrival*. The first customer inserted on a route is referred to as the *seed customer*. Once the seed customer has been identified and inserted, the sequential insertion heuristic algorithm considers, for the unrouted nodes, the insertion place that minimizes a weighted average of the additional distance and time needed to include a customer in the current partially constructed route. This second step is referred to as the *insertion criteria*, and involves savings criteria introduced by Dullaert *et al* [3] and Golden *et al* [4]. The third step, the *selection criteria*, tries to maximize the benefit derived from inserting a customer in the current partial route rather than on a new direct route. Note that the terms *nodes* and *customers* are used interchangeably.

A shortcoming of the sequential insertion heuristic is that it considers all unrouted nodes when calculating the insertion and selection criteria during each iteration. The fact that *all* unrouted nodes are considered makes it computationally expensive. The occurrence of obvious infeasible nodes in a partially constructed route therefore becomes significant. The concept of *time window compatibility* has first been introduced by Joubert and van Schalkwyk [7]. A matrix, referred to as the *Time Window Compatibility Matrix* (TWCM), is used as the mechanism to

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

calculate the compatibility between all nodes in the network. The introduction of the time window compatibility concept assists in identifying, and eliminating, the obvious infeasible nodes, resulting in a more effective and robust route construction heuristic.

The purpose of time window compatibility is to determine the time overlap of all edges, or node combinations, (i, j), where $i, j \in \{0, 1, 2, \ldots, N\}$, and N the total number of nodes in the network. During the route construction phase, time window compatibility can be checked, and obvious infeasible nodes can be eliminated from the set of considered nodes. The TWCM is a non-symmetrical matrix as the sequence of two consecutive nodes, i and j, is critical.

Let:

N	be the total number of nodes
e_i	be the earliest allowed arrival time at customer i, where $i = \{0, 1, \ldots, N\}$
l_i	be the latest allowed arrival time at customer i, where $i = \{0, 1, \ldots, N\}$
s_i	be the service time at node i, where $i = \{0, 1, \ldots, N\}$
t_{ij}	be the travel time from node i to node j, where $i, j = \{0, 1, \ldots, N\}$
$a_j^{e_i}$	be the actual arrival time at node j, given that node j is visited directly after node i, and that the actual arrival time at node i was e_i, where $i, j = \{0, 1, \ldots, N\}$
$a_j^{l_i}$	be the actual arrival time at node j, given that node j is visited directly after node i, and that the actual arrival time at node i was l_i, where $i, j = \{0, 1, \ldots, N\}$
TWC_{ij}	be the time window compatibility when node i is directly followed by node j

The generalized equation (1) is proposed and addresses five distinct time window scenarios.

$$TWC_{ij} = \begin{cases} \min\{a_j^{l_i}, l_j\} - \max\{a_j^{e_i}, e_j\} & \text{if } l_j - a_j^{e_i} > 0 \\ -\infty & \text{otherwise} \end{cases} \tag{1}$$

The interested reader is referred to Joubert [6] for a detailed discussion on each scenario, and appropriate examples.

The time window compatibility matrix, TWCM, is calculated before the route building heuristic is evoked. In each iteration of the sequential insertion heuristic, Solomon [11] calculates the insertion and selection criteria for each edge on the partially constructed route, irrespective of the compatibility of the time window of the node considered for insertion with the time windows of the two nodes forming

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

the edge. This paper presents an improved case. Consider the example where node u is considered for insertion between nodes i and j. As the TWCM is already calculated, it is possible to check the compatibility of node u with the routed nodes i and j. If either TWC_{iu} or TWC_{uj} is negative infinity ($-\infty$), indicating an incompatible time window, the insertion heuristic moves on and considers the next edge, without wasting computational effort on calculating the insertion and selection criteria.

As opposed to the two most common initialization criteria, namely *customer with earliest deadline*, and *furthest customer*, as suggested by Dullaert *et al.* [3], this paper proposes the use of the TWCM to identify seed nodes based on their time window compatibility. The node with the most infeasible instances (either as origin or destination), are identified as the seed customer. Ties are broken arbitrarily. It may be possible to not have any infeasible time window instances. In such a scenario, a *total compatibility* value can be determined for each node a, and is calculated as

$$\sum_{i=1, i\neq a}^{M} TWC_{ia} + \sum_{j=1, j\neq a}^{M} TWC_{aj} + TWC_{aa} \tag{2}$$

where M refers to all the unrouted nodes, including all instances of those nodes that are split artificially. The customer with the lowest total compatibility is selected as seed customer.

3 Initial solution algorithm

The algorithm uses both vehicle and customer information as input. Customers with multiple time windows are artificially split: a customer with t time windows are split into t artificial customers, each with a single time window. When any one customer is routed, the algorithm eliminates its related artificial customers.

A vehicle is assigned to a *tour*, while each tour can be made up of one or more *routes* (referred to as double scheduling). The algorithm check for multi-route feasibility by calculating the arrival time at the depot at the end of the current route. If the arrival time, plus a specified buffer, does not exceed the time window for the depot (operating hours), a new route for the current tour is initiated. Otherwise, a new tour is established.

4 Results

Solomon [11] discusses the generation of data sets for the *Vehicle routing and scheduling problems with time window constraints* (VRPSTW), and indicates that the design of these data sets highlight several factors that affects the behavior of his routing and scheduling heuristics. The corresponding six data sets, referred to as $R1$, $R2$, $C1$, $C2$, $RC1$, and $RC2$, are often used and referred to in literature.

The data used for the customer coordinates and demands are based on the work of Christofides *et al.* [1], and are classified into one of three categories:

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

- Randomly distributed customers (denoted by an R prefix)
- Clustered customers (denoted by a C prefix)
- Semi-clustered customers (denoted by an RC prefix)

By semi-clustered is implied a random mix of both randomly distributed and clustered customers.

The length of the route-time is regarded as a capacity constraint and, along with the vehicle capacities, limit the number of customers serviced by a specific vehicle. The short scheduling horizon problems are denoted by a "1" as a suffix. The problems denoted by a "2" suffix, on the other hand, have a large scheduling horizon, and along with the vehicle capacities, allow a larger number of customers to be serviced by a single vehicle.

Solomon's data sets have become a benchmark for vehicle routing problem variants, although the sets are often used with some modification due to the specific variant's particularity. The published data sets [12] assume a homogeneous fleet, indicate a given vehicle capacity, and assume infinite availability of vehicles. Furthermore the sets only indicate a single time window for each geographically distributed customer.

The aim of the algorithm is to test the feasibility of integrating multiple time windows, a heterogeneous fleet, and double scheduling into an initial solution heuristic. As none of these three specific variants are addressed by the Solomon data sets, it was necessary to generate a unique data set to accommodate, and integrate the problematic variants, yet still have resemblance to the familiar benchmarks in literature. The problem of developing an appropriate data set meant addressing both multiple time windows, and a heterogeneous fleet. Double scheduling is a function of the algorithm, and does not require any manipulation of the data sets.

It originally seemed appropriate to use two of the Solomon sets for a specific class of problem, each with 100 customers and a single, unique time window, to create a new set of 100 customers with multiple time windows. It turned out to be futile, as the time windows for different data sets of the same classification, as presented by Solomon [12], had very similar, if not exactly the same, time windows. The extended data sets presented by Homberger [5] are developed in the same manner as Solomon's sets, but have sets with 200, 400, 600, 800, and 1000 customers.

For each of the six problem classes, an extended set with 200 customers from Homberger's sets is used to create a new set with 100 customers, but with two time windows. Table 1 illustrates an excerpt from the data used to create a new set for the $C1$ class. In the Homberger data sets, the *customer number*, *x-coordinate*, *y-coordinate*, *demand*, *service time*, and the *start-* and *end times* of a single time window, are given.

Table 1 indicates the case where the single time windows of customers 101 through 200 have been used as *second* time windows for customers 1 through 100. For example, the time window for customer 4 is given as $(616, 661)$, measured in minutes from 0:00am. Customer 104's time window is given as $(128, 195)$ in the original data set. Customer 104's time window now becomes the second time

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

window for customer 4, while all other data about customers 101 through 200 is disregarded. Observe, however, that the two time windows specified for customer 5 overlap, and do not yield two unique time windows as is the case for customer 4.

Table 1: Constructing data sets with multiple time windows.

Customer				**Time window 1**		**Time window 2**	
(i)	**x**	**y**	**...**	e_i^1	l_i^1	e_i^2	l_i^2
⋮							
4	4	28	...	616	661	128	195
5	25	26		128	179	142	197
6	86	37		478	531	754	814
7	1	109	...	616	680	583	647
⋮							

Procedure 1 indicates the procedure used to manipulate the time windows from multiple sets into a single data set. Where time windows overlap, the new, single time window, is defined to start at the opening of the earlier time window, and end at the closing of the later time window. After the manipulation of the data, the start of the first time window for customer i is denoted by E_i^1, and the end of the time window by L_i^1. Where only one time window exist, no second time window is specified. Alternatively, the start of the second time window for customer i is denoted by E_i^2, and the end of the time window by L_i^2.

Procedure 1 Creating a data set with multiple time windows

if either $e_i^1 > l_i^2$ or $e_i^2 > l_i^1$ **then**
  Number of time windows is 2
  $E_i^1 = \min\{e_i^1, e_i^2\}$
  $L_i^1 = \min\{l_i^1, l_i^2\}$
  $E_i^2 = \max\{e_i^1, e_i^2\}$
  $L_i^2 = \max\{l_i^1, l_i^2\}$
else
  Number of time windows is 1
  $E_i^1 = \min\{e_i^1, e_i^2\}$
  $L_i^1 = \max\{l_i^1, l_i^2\}$
end if

Liu and Shen [9, 10] propose a specific fleet structure with the introduction of their insertion-based savings heuristic for a heterogeneous fleet. The proposed cost structure sees the cost of a vehicle more than doubles when its capacity doubles. It was considered appropriate to use the given fleet composition, as Dullaert *et al.* [3] challenges the cost structure presented, but did not propose a new cost structure.

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

Table 2: Solution summary.

Test set	Without *TWC*			With *TWC*			Distance saving	Time saving
	Number of tours	Number of routes	Total distance	Number of tours	Number of routes	Total distance		
*C*1	11	40	8,914 km	11	22	7,330 km	18%	51%
*C*2	4	19	7,246 km	4	6	7,240 km	0%	39%
*R*1	8	70	11,172 km	18	72	11,378 km	-2%	-7%
*R*2	2	13	9,990 km	5	7	8,722 km	13%	61%
*RC*1	6	50	9,026 km	26	48	8,706 km	4%	25%
*RC*2	2	21	8,622 km	12	19	7,748 km	10%	42%

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)

It is important to evaluate the contribution that the proposed *time window compatibility* has on the results of the algorithm. For this purpose, a comparative *control algorithm* is created. The control algorithm differs only in two respects from the proposed algorithm:

- It does not evaluate nodes for *time window compatibility* when calculating the insertion criteria, and therefore considers every node for insertion on every edge of a partially constructed route.
- As no *time window compatibility* is calculated for any node, the initialization criteria is changed to identify the seed customer as the unrouted customer with the earliest deadline.

Detailed figures that illustrates the geographical distribution of customers for each of the six data sets, as well as a graphical performance comparison can be requested from the author. The results are summarized in Table 2.

5 Discussion and implications

The main objective of the proposed algorithm is to improve the computational efficiency of the initial solution heuristic. The algorithm has improved efficiency by 35% on average, with improvements for all data sets except for a slight derogating impact for randomly distributed customers with a short scheduling horizon ($R1$).

Scheduling distances have been improved by 7% on average, with a slight increase again for customers in the $R1$ set.

There is not a clear relationship between the performance of the proposed algorithm and the geographical and scheduling horizon characteristics of the data sets. The improvements achieved and presented in this paper is significant, and contributes to the scholarship of vehicle routing and scheduling. An improvement in the quality of the initial solution also sets the agenda for the incorporation of the time window compatibility concept into improvement metaheuristics such as the Tabu Search.

References

[1] N. Christofides, A. Mingozzi, and P. Toth. *The Vehicle Routing Problem*. Combinatorial Optimizations. John Wiley & Sons, New York, 1979.

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

[3] W. Dullaert, G.K. Janssens, K. Sörensen, and B. Vernimmen. New heuristics for the fleet size and mix vehicle routing problem with time windows. In 9^{th} *World Conference on Transport Research*, COEX Convention Center, Seoul, July 2001.

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

[5] J. Homberger. Extended SOLOMON's VRPTW instances. World wide web

at *http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm*, September 2003.

[6] J.W. Joubert. An initial solution heuristic for the vehicle routing and scheduling problem. Master's thesis, Industrial Engineering, University of Pretoria, South Africa, November 2003.

[7] J.W. Joubert and W.T. van Schalkwyk. Time window compatibility for enhanced initial solutions. 2003.

[8] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:345–358, 1992.

[9] F.-H. Liu and S.-Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, 50:721–732, 1999.

[10] F.-H. Liu and S.-Y. Shen. A method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows. *Proceedings of the National Science Council, Republic of China, ROC(A)*, 23(4):526–536, 1999.

[11] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time windows. *Operations Research*, 35(2):254–265, 1987.

[12] M.M. Solomon. VRPTW benchmark problems. World wide web at *http://web.cba.neu.edu/ msolomon/problems.htm*, June 2003.

[13] K.C. Tan, L.H. Lee, Q.L. Zhu, and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15:281–295, 2001.

[14] A. Van Breedam. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers and Operations Research*, 28:289–315, 2001.

Urban Transport X, C. A. Brebbia & L. C. Wadhwa (Editors)