Journal of the Operational Research Society (2001) 52, 523–537 0160-5682/01 $15.00

www.palgrave.com/jors

A greedy look-ahead heuristic for the vehicle routing problem with time windows

G Ioannou*, M Kritikos and G Prastacos

Athens University of Economics and Business, Athens, Greece

In this paper we consider the problem of physically distributing finished goods from a central facility to geographically dispersed customers, which pose daily demands for items produced in the facility and act as sales points for consumers. The management of the facility is responsible for satisfying all demand, and promises deliveries to the customers within fixed time intervals that represent the earliest and latest times during the day that a delivery can take place. We formulate a comprehensive mathematical model to capture all aspects of the problem, and incorporate in the model all critical practical concerns such as vehicle capacity, delivery time intervals and all relevant costs. The model, which is a case of the vehicle routing problem with time windows, is solved using a new heuristic technique. Our solution method, which is based upon Atkinson's greedy look-ahead heuristic, enhances traditional vehicle routing approaches, and provides surprisingly good performance results with respect to a set of standard test problems from the literature. The approach is used to determine the vehicle fleet size and the daily route of each vehicle in an industrial example from the food industry. This actual problem, with approximately two thousand customers, is presented and solved by our heuristic, using an interface to a Geographical Information System to determine inter-customer and depot–customer distances. The results indicate that the method is well suited for determining the required number of vehicles and the delivery schedules on a daily basis, in real life applications.

Keywords: vehicle routing; heuristics; logistics;

Introduction

On-time production and timely distribution of finished goods is an intense problem faced by most major industries in every country. Product proliferation, short product life and constant thrive for expanded market share define a severely competitive business environment that places heavy constraints and requirements on the producers, who are also responsible, in the majority of the cases, for the delivery of products to the final sales point.[1,2] In this paper we address the back-end problem of the physical distribution of goods to customers' locations, after all production and packaging stages have been completed. The problem is of particular importance since the relevant costs (acquisition of delivery vehicles and their operation, ie labour and cost of consumables) constitute a large proportion of the overall investment and operational costs of a producer.[3] Furthermore, the ability to meet customer demand on time is directly dependent on the accuracy of delivery schedules, and customer satisfaction is defined by-and-large through the reliability of the distribution network.[4] Finally, shelf space (a critical performance measure for some industrial sectors such as food and beverage production and distribution) can be maintained only via daily deliveries, mainly due to the short shelf life of the products, thus emphasising the importance of the back-end distribution problem.

The industrial situation we address in this paper can be accurately described as follows. Consider a central facility (or warehouse) that produces (consolidates, respectively) several items that have to be delivered in relatively short times to a set of customers. The latter are geographically dispersed within a distance radius that allows for demand to be satisfied through daily deliveries. We assume that customer demand is known when a delivery schedule is determined, as is the distance and travel time between the facility (warehouse) and each customer's location, as well as between each pair of customers' locations. In addition, the time interval during which the delivery has to take place is also known (fixed when sales are negotiated and finalised by the Sales and Marketing department). This interval is bounded by the earliest and latest time of the day that the delivery to a particular customer has to be completed. The daily delivery process is performed according to the following steps. Products are loaded on appropriate vehicles at the facility (warehouse) and, subsequently, they are transported via a road network to the customers' locations. At each location, quantities that equal customer demand are unloaded from the vehicle, paper work (shipping documents, various bills and invoices) is filled in and exchanged, and then vehicles travel to subsequent customers' locations where the process is repeated. After all

*Correspondence: G Ioannou, Graduate Program in Decision Sciences, 8th Floor, 47A Evelpidon Street and 33 Lefkados Street, Athens 113-62, Greece. E-mail: ioannou@aueb.gr

deliveries have been performed (end of a shift), vehicles return to the facility (warehouse) for the following daily cycle.

The problem is a case of the well known vehicle routing problem with time windows (VRPTW).[5] The depot of this VRPTW is the shipping area of the facility (or warehouse), including the docks for truck loading. Vehicles are loaded at the depot up to (or sometimes below, according to customer requests) their capacity, and subsequently, they perform closed tours during which they visit several customers and deliver products (usually in pallets or boxes) that equal each customer's demand. The depot represents the originating and terminating node of all tours; the latter are equivalent to the routes of each vehicle in the classical VRPTW. The goal is to determine the minimum number of vehicles that are required to service all customers with the minimum operating cost, ie in the minimum time or following the paths of minimum distance. This combined objective reflects the trade-off between fixed acquisition cost of vehicles and variable transportation cost of distribution.

Due to the importance of the problem, there is a vast literature that addresses modelling and solution aspects of the VRPTW. Bodin *et al*[6] presented a comprehensive overview of the original vehicle routing problem, ie the problem without time windows, and reviewed the published work up to the 1980s, related to algorithmic developments. Desrochers *et al*[7] developed a mathematical formulation of the VRPTW, surveyed the routing and scheduling problems with time windows, highlighted the most significant elements of the existing solution methodologies, and provided an analysis of the mechanics of some key solution methods. The computational complexity of the VRPTW has also been thoroughly examined, and the problem has been proven to be NP-hard.[8] This, in turn, has led the majority of researchers to focus on heuristic and meta-heuristic methods (such as tabu search, simulated annealing, and genetic algorithms) to derive approximate solutions of acceptable quality in reasonable computational time. The review papers of Gendreau *et al*[5] and Laporte[9] provide all necessary pointers to the research efforts in the area of solution methods developed for the VRPTW in the 1980s and 1990s.

A key reference concerning our work is the greedy construction heuristic proposed by Atkinson[1] that incorporates a look-ahead capability by examining the effect that the selection of a customer for service may have on other customers, not routed yet. This method was first introduced in a vehicle-scheduling case study,[10] was extended to a broad, general class of combinatorial optimisation problems, including the VRPTW,[1] and was further developed to incorporate randomisation and aspects of artificial intelligence.[11] The approach provides a powerful additional capability to the route construction mechanics that has a significantly positive impact on the overall solution performance-quality. In particular, at each stage of the proposed method, a customer is selected for insertion into the route under construction by minimising a function that combines, through appropriate weights, travel times and distances and time window information. By doing so, it achieves small inter-customer distances within the routes and retains flexibility for 'subsequent' customers to meet their own time window constraints.

From the literature survey papers, we conclude that the approaches developed for the vehicle routing problem with time windows can be classified as follows: (a) improvement and construction heuristics; most of these methods result in relatively mediocre-to-good solutions in short computational times; (b) meta-heuristics; these methods result in higher quality solutions, but in relatively large computational times (compared to heuristic methods) for practical applications, where decisions for vehicle routes with hundreds-to-thousands of customers have to be determined in minutes. Furthermore, implicit enumeration schemes have also been developed for the VRPTW. For example, methods based on Lagrangian relaxation,[12,13] column generation,[14] or typical branch and bound approaches,[15] have been reported. Such optimal methods, although capable of solving problems with up to 100 customers for some VRPTW instances, cannot be applied to practical distribution problems, where the number of customers is of the order of thousands and decision-makers require answers in very short times (minutes before the start of a shift).

New heuristics are, therefore, necessary to achieve solutions of quality comparable to those of meta-heuristics, in very short computational time, in order to be applicable to actual industrial cases. In this paper, we present a new such heuristic for the VRPTW. The heuristic is 'intuitively intelligent' since it examines all time windows-related information from the model and the partial solution developed at each construction stage. This 'look-ahead' property, which we adopt from Atkinson[1] and enhance with the new criteria for next-customer selection and in-route customer insertion, represents the major contribution of our work. The application of the heuristic to benchmark problems leads to an improvement upon the results from the literature. We consider the best-published results both from heuristics and meta-heuristics; our approach outperforms all existing heuristics, and provides results comparable to those of meta-heuristics in computational times of at least an order of magnitude shorter. Furthermore, via the new heuristic we are capable of solving an actual industrial problem with almost two thousand customers and substantially improve upon the current practice of the company.

The remainder of the paper is organised as follows. First we present a comprehensive formulation of the vehicle routing problem with time windows, as applied to the distribution of finished goods. Then, we describe extensively the attributes and mechanics of the heuristic we have developed. Subsequently, we provide a thorough analysis of the computational results obtained by applying the new heuristic on several data sets from the literature (small to medium sizes) and from the Web (large problem instances). Finally, we report on a new large data set from an industrial case study, and conclude the paper.

Model formulation

The VRPTW can be stated as follows. Find a set of closed routes, for a fleet of $|V|$ identical vehicles (V = set of available vehicles, ie the maximum number of vehicles that can be used for deliveries) with known capacity C, servicing a set of customers, from a central depot at minimum cost. The number of customers is $n-1$, ie $|L|-1=n-1$, where L is the set of customers including the depot, which is a distinct node of the underlying connected graph. Indices i, j and u refer to customers and take values between 2 and n, while index $i=1$ refers to the depot; an additional index k counts the vehicles ($k=1,\ldots,|V|$). Vehicles are initially located at the central depot. Each customer i poses a demand q_i, and is bounded by a time window $[e_i, l_i]$ that models the earliest and latest time that customer i can be serviced by a vehicle. Furthermore, a service time, s_i, is required for each customer i. Each vehicle route originates and terminates at the central depot, while each customer is serviced by exactly one vehicle. There is a cost c_{ij}, a travel time t_{ij} and a distance d_{ij} associated with the path from customer i to customer j. For simplicity, we assume that c_{ij}, t_{ij} and d_{ij}, are equivalent measures with appropriate adjustments, and we use these notations interchangeably in the remainder of the paper. Furthermore, a cost w_k is relevant to the activation of vehicle $k \in V$; this is a one-time cost and it is related to the fixed costs for the acquisition of vehicle k. An additional view of this cost can be the set-up cost for a vehicle on a daily basis; appropriate normalisation with respect to the variable costs c_{ij} is necessary in both cases. Each route must satisfy capacity and time window constraints. Capacity constraints state that the total quantity of goods delivered cannot exceed the vehicle capacity C. On the other hand, time window constraints impose the following conditions: a vehicle cannot service customer i before the lower bound e_i and after the upper bound l_i limits of the customer's time window. However, the vehicle can arrive before the time window lower bound and wait until the allowable service time begins.

The mathematical programming formulation of the VRPTW requires three groups of variables. The first group models the sequence in which vehicles visit customers, and is defined as follows:

$$x_{ij}^k = \begin{cases} 1 & \text{if customer } i \text{ follows customer } j \text{ in the sequence of customers visited by vehicle } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The second group of variables, denoted by a_i and p_i for each customer i, specifies the arrival–departure time to/from customer i, respectively. Finally, z_k is a binary variable defined as follows:

$$z_k = \begin{cases} 1 & \text{if vehicle } k \text{ is active} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Note that a vehicle is active when it services at least one customer. Given the above-defined variables and parameters, the problem can be formulated as follows:

Problem VRPTW

$$\text{Minimise} \quad \sum_{k=1}^{|V|}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}^k + \sum_{k=1}^{|V|} w_k z_k \tag{3}$$

Subject to:

$$\sum_{i=1}^{n}\sum_{k=1}^{|V|} x_{ij}^k = 1 \qquad \forall\, j=2,3,\ldots,n \tag{4}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{|V|} x_{ij}^k = 1 \qquad \forall\, i=2,3,\ldots,n \tag{5}$$

$$x_{ij}^k \leqslant z_k \qquad \forall\, i,j=1,2,\ldots,n \tag{6}$$

$$\sum_{j=2}^{n} x_{1j}^k \leqslant 1 \qquad \forall\, k=1,2,\ldots,|V| \tag{7}$$

$$\sum_{i=2}^{n} x_{i1}^k \leqslant 1 \qquad \forall\, k=1,2,\ldots,|V| \tag{8}$$

$$\sum_{i=2}^{n} x_{iu}^k - \sum_{j=2} x_{uj}^k = 0 \quad \forall\, k=1,\ldots,|V|, \quad \forall\, u=1,\ldots,n \tag{9}$$

$$\sum_{i\in F}\sum_{j\in F} x_{ij}^k \leqslant \sum_{i\in F}\sum_{j\in L} x_{ij}^k - 1 \quad \forall F \subseteq L: 2 \leqslant |F| \leqslant \sum_{i\in L}\sum_{j\in L} x_{ij}^k, \forall\, k \in V \tag{10}$$

$$\sum_{i=1}^{n} q_i\left(\sum_{j=1}^{n} x_{ij}^k\right) \leqslant C \qquad \forall\, k=1,2,\ldots,|V| \tag{11}$$

$$a_j \geqslant (p_i + t_{ij}) - (1 - x_{ij}^k)M \qquad \forall\, i,j=1,2,\ldots,n, \quad \forall\, k=1,2,\ldots,|V| \tag{12}$$

$$a_j \leqslant (p_i + t_{ij}) + (1 - x_{ij}^k)M \qquad \forall\, i,j=1,2,\ldots,n, \quad \forall\, k=1,2,\ldots,|V| \tag{13}$$

$$a_i \leqslant p_i - s_i \qquad \forall\, i = 1, \ldots, n \tag{14}$$

$$e_i \leqslant p_i \leqslant l_i \qquad \forall\, i = 1, \ldots, n \tag{15}$$

$$a_1 = 0 \tag{16}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall\, i, j = 1, \ldots, n, \quad \forall\, k = 1, 2, \ldots, |V| \tag{17}$$

$$z_k \in \{0, 1\} \qquad \forall\, k = 1, 2, \ldots, |V| \tag{18}$$

The objective function (3) models the trade-off between route and vehicle costs. The function couples the operation of the vehicles with their acquisition or initial set-up. The first term of (3) reflects the cost of the routes followed by all vehicles after they depart from and before they return to the depot, as well as the cost of the first and last segment of each route. The second term of (3) reflects the total vehicle acquisition or set-up cost.

Constraints (4) and (5) ensure that exactly one vehicle enters and departs from every customer and from the depot. Constraint (6) relates the x and z variables, guaranteeing that no customers are services by inactive vehicles, ie by vehicles with $z_k = 0$. Constraints (7) and (8) account for the availability of vehicles by bounding the number of arcs, related to each vehicle k directly leaving from and returning to the depot, to below one, respectively. Constraint set (9) is the typical flow conservation equation that ensures the continuity of each vehicle route. Constraint (10) eliminates sub-tours; this is typical in all vehicle routing and travelling salesman problem formulations (see, eg Golden and Assad[3]). Constraint (11) imposes an upper bound equal to the vehicle capacity C to the total customer demand that is loaded on each of the vehicles. Constraints (12)–(16) are related to time windows and guarantee the feasibility of the schedule for each vehicle. In particular, constraints (12) and (13) ensure that, if customers i and j are consecutive in the schedule of vehicle k, then the arrival time at customer j equals the departure time from customer i, plus the travel time between these two customers. Note that M is a large number, a typical 'trick' in mathematical programming formulations to eliminate 0–1 infeasibilities. In case customers i and j are not serviced by the same vehicle or are not consecutive, constraints (12) and (13) are inactive. Constraints (14) and (15) guarantee that the relationships between arrival time, departure time, and service time with respect to customer i are compatible to the customer's time window. Constraint (16) sets the departure time from the depot equal to zero, since all routes originate at the depot. The last two sets of constraint, (17) and (18), enforce integrality for the x_{ij}^k and z_k variables, respectively.

The model of (3)–(18) is a typical binary program that mathematically depicts the vehicle routing problem with time windows. In the model we have incorporated both one-time and recurring costs, and we have captured all critical factors related to time windows. There exists no special need to constrain the total time–distance of each vehicle route. Assuming that there is a time window associated with the depot, constraints (12)–(16) impose such restrictions. Several authors have proposed various versions of the model (see, eg, Chiang and Russell[16]). However, the model of (3)–(18) best fits our delivery-distribution problem. It is a complex combinatorial optimisation model that requires substantial computational effort for determining approximate solutions even for medium size problems. Due to the problem's computational complexity, heuristic or meta-heuristic methods are appropriate for obtaining solutions of relatively good quality in reasonable computational times. Optimal solutions can only be achieved by implicit enumeration schemes; such solutions have been reported in ref [14], and are used later in the paper for evaluating the method we have developed.

Solution algorithm

Given the model formulated in the previous section, we now proceed to the development of a solution approach to tackle effectively and efficiently the VRPTW. The proposed heuristic can be classified as a route-construction sequential approach[17] since it builds vehicle routes one at a time. As a solution framework, we adapt the generic insertion framework proposed by Solomon[18] and used by other authors.[19] According to this framework, after initialising a route with a 'seed' customer, our heuristic uses two criteria to insert a new customer between two adjacent customers in the current partial route. The first criterion identifies the best customer to be inserted in the current route, while the second criterion determines the best insertion place in the current route. The procedure is repeated until no further customer can be inserted into the current route. Then, a new 'seed' customer is identified from the non-routed ones and a new route is initiated. The overall loop is performed until all customers have been assigned to vehicles. The solution procedure is straightforward with very simple mechanics. However, it is based on new criteria for customer selection and insertion, which are motivated by the minimisation function of the greedy look-ahead solution approach of Atkinson.[1] We utilise all time windows related information and attempt to expand the set of customers that are feasible for route insertion at each stage of the algorithm. In this way, we expect to derive high quality solutions by exploiting the time window constraints.

Intuition—motivation

The vehicle routing problem with time windows can be decomposed into a bin-packing problem (or a set-partitioning problem) and a scheduling problem (or travelling salesman problem), with some form of precedence constraints to account for time window constraints. The complexity of the VRPTW emanates from the inherent complexity of its underlying constituent elements. One can solve with success any of the two underlying sub-problems of the VRPTW; nevertheless, combining such solutions and obtaining vehicle routes that satisfy all time windows is still an NP-hard problem.[20] Thus, a key determinant factor for obtaining good solutions for vehicle routing problems with tight time window constraints is the effective utilisation of time window constraints themselves in the solution approach. What complicates matters further is the strong relationship between customers' time availability for service; these relationships are introduced through the time windows that constrain the sequence in which customers are visited by vehicles. Thus, to ensure feasibility to the VRPTW, an effective solution method should account for all these relationships and safeguard (see also Atkinson[1] for a comprehensive discussion of the points below):

(a) Short inter-customer distances (times).
(b) Minimal vehicle activation.
(c) Sequential selection of customers that minimally restricts the solution space, in the case of construction heuristics. Note that the restriction to the solution space occurs when a customer assignment to a vehicle (thus, to a sequence of customers) is fixed without accounting for the effect of such an assignment on all time window constraints (look-ahead property[1]).

Most of the approaches developed to date do not explicitly consider such characteristics when attempting to solve the VRPTW. In particular, the majority of the existing heuristics, either: (i) myopically select 'closest' customers, or (ii) create simple interaction metrics that do not handle the inter-time-window intrinsic relationships.

Motivated by this lack of an in-depth consideration of inter-customer relationships generated by time windows, we develop new criteria for customer selection and route insertion (according to Atkinson's[1] greedy look-ahead optimisation framework) and implement these criteria in a new solution method. The basic idea behind the criteria and their solution implementation is that a customer u selected for insertion into a route should minimise the impact of the insertion on the customers already in the route under construction, on all non-routed (not assigned to vehicles) customers, and on the time window of customer u himself. In order to measure these impacts, which are captured by the term 'look-ahead property', we define three new criteria, ie:

(a) IS_u = own impact, ie relationship between arrival time to customer u (a_u) and the lower bound on the service time of u (e_u).
(b) IU_u = external impact, ie the impact of customer's u insertion on non-routed customers.
(c) IR_u = internal impact, ie the impact of customer's u insertion on customers already routed within the route under construction.

Criteria (a), (b) and (c) are based on time and distance increases of scheduling measures, and are discussed in detail below.

Customer selection and route insertion criteria

The impact of inserting customer u in the route under construction refers to the closeness to the earliest service time (e_u) of the vehicle arrival time to customer u (a_u). This depends on the location of customer u's insertion and is given by

$$IS_u = a_u - e_u \tag{19}$$

IS_u is a measure of the coverage of the selected customer's time window, which results from the insertion of u into the current route. A value of IS_u close to zero provides slack for the insertion of additional customers before and after u, as the route construction algorithm proceeds. This decreases the possibility of customer infeasibility for insertion to the route. Consequently, a near-zero value for IS_u expands the set of customers that can be selected for insertion to the current route as the route construction process evolves.

To illustrate the second criterion (external impact) we consider customers u and j, as in Figure 1. Customer u is a non-routed customer selected for insertion on the current route, and customer j is another non-routed customer. The time windows of these customers are depicted in Figure 1. After scheduling a customer u, the problem of selecting a new one for route insertion becomes harder; this is due to the overlapping of the time windows of non-routed customers. A necessary condition for a vehicle to visit customer j after the selected for insertion customer u is (assuming feasibility)

$$e_u + d_{uj} \leqslant l_j \vee e_j + d_{ju} \leqslant l_u \tag{20}$$

In (20) we consider both the case shown in Figure 1, ie customer u preceding customer j, and the reverse. It is obvious that the insertion of customer u, which minimises the non-negative difference of $[l_j - (e_u + d_{uj})]$ and $[l_u - (e_j + d_{ju})]$, is expected to be a good selection. This is because minimising this difference decreases the affected component of the time window of non-routed customer j due to the insertion of customer u into the current route. Note that, depending on the precedence relationship between customers j and u, one of the two differences is non-negative.

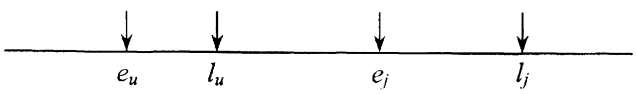

Figure 1 Time windows of customers u and j.

The metric of external impact calculates the real coverage of time windows of the non-routed customers resulting from the insertion of the selected customer into a route. Examining all possible cases, we can formulate the criterion as follows:

$$IU_u = \sum_{j \in J-\{u\}} \{[1/(|J|-1)] \times max\ \{(l_j - e_u - d_{uj}), (l_u - e_j - d_{uj})\} \quad (21)$$

In (21), J is the set of the non-routed customers at any stage of the route construction procedure. The metric of (21) formally expresses the average unutilised time window over all non-routed customers.

To define the internal impact, we require three metrics: local and global disturbance resulting from the insertion of a selected customer into the current route and accessibility of this customer to the current route. We can define these measures as follows. Let u be the customer selected to be inserted between customers i, j of the current route. This insertion causes a distance increase:

$$c_{1u}(i, j) = d_{iu} + d_{uj} - d_{ij} \quad (22)$$

In addition, a time delay related to the vehicle arrival, C_{2u}, is introduced at customer j. Figure 2 illustrates this time delay, which is the marginal time feasibility of customer u.[18] It is defined as the time difference between the vehicle arrival time at customer j, before and after the insertion of customer u into the current route. Equation (23) provides the mathematical expression for C_{2u}.

$$c_{2u}(i, j) = [l_j - (a_i + s_i + d_{ij})] - [l_j - (a_u + s_u + d_{uj})] \quad (23)$$

The marginal time feasibility calculates the urgency of the routed customer j. It contains Solomon's measure of vehicle delay at customer j due to the insertion of the customer u into the current route.

Finally, the insertion of customer u between customers i and j defines a time gap, c_{3u}, between the latest service time l_u of the customer u and the time of the vehicle arrival at customer u. This measure expresses the compatibility of the time window of the selected customer with the specific insertion place in the current route. The formula for this metric is

$$c_{3u}(i, j) = l_u - (a_i + s_i + d_{iu}) \quad (24)$$

It is evident that a non-routed customer u that has minimum time gap should be inserted into the route.

The best insertion position for a non-routed customer is determined by a weighted combination of the above-defined criteria, ie of the extra distance, the marginal time feasibility and the time window compatibility. Mathematically, we can define this criterion, which we call local disturbance, LD_u, of customer u's insertion between customers i and j as follows:

$$LD_u(i, j) = b_1 c_{1u} + b_2 c_{2u} + b_3 c_{3u} \quad (25)$$

In (25) $b_1 + b_2 + b_3 = 1$, and b_1, b_2 and $b_3 \geqslant 0$.

For the current route r and for the non-routed feasible customer u, we define the global disturbance IS_u as the summation of the local disturbances LD_u for every insertion point into the route:

$$IR_u = \sum_{(i,j) \in I_r} LD_u/|I_r| \quad (26)$$

In (26), I_r is the set of all feasible insertion places of customer u into route r, and $|I_r|$ is the cardinality of this set. Note that the elements of I_r are ordered pairs of customers (i, j) that belong to the current route. The inverse of the average of the local disturbances defines the accessibility, ie:

$$ACC_u = (IR_u)^{-1} \quad (27)$$

The metric of the accessibility expresses the internal impact of the selected customer u on the current route, since a customer with large accessibility causes small internal impact on the current route. This is a combined result from all of the three sub-metrics that form the accessibility. Our heuristic approach requires that customers selected for insertion should have the best accessibility to the current route.

Now we can define the customer selection criterion, which accounts for all types of impacts described above. We use a simple linear relationship similar to (25) to merge the effects of internal, external and own impact for non-routed customer u:

$$Impact(u) = b_s IS_u + b_e IU_u + b_r IR_u \quad (28)$$

In (28) $b_s + b_e + b_r = 1$, and b_s, b_e, and $b_r \geqslant 0$. Note that the weights b_s, b_e, and b_r define the relative contribution of each individual metric (internal, external and own impact) to the overall selection criterion. Their values could be defined after statistical experimentation.

The composite customer selection criterion of (28) ensures that a customer u selected for insertion into a route will minimise the impact of the insertion on the customers already in the route under construction, on all non-routed customers, and on himself. Such a criterion should lead a greedy method to good solutions since it will

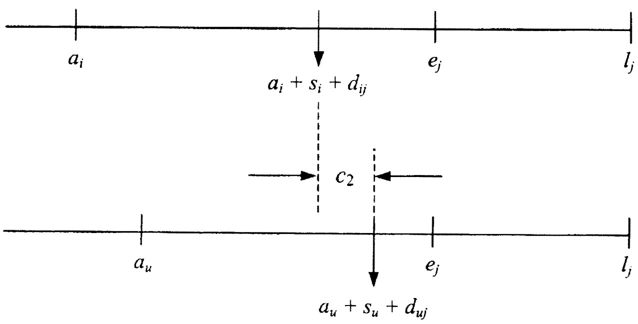

Figure 2 Marginal time feasibility.

allow the exploitation of a large solution space and will account for the inter-customer relationships through the time windows. We expect the criterion by itself to lead an unsophisticated method to solutions of high quality, and such expectations are proven by the application of the IMPACT heuristic we develop and test in the remainder of the paper. Furthermore, (28) incorporates the three additional parameters related to $LD_u(i, j)$, ie b_1, b_2, and b_3. Consequently, *Impact*(u) depends upon a 3×3 parameter combination, and adjustment/fine tuning of these parameters is necessary to achieve the best possible approximate solutions. This parameter fine tuning should be performed off-line in real-life situations, and the results should be implemented in the solution procedure. An example for the calculation of the various measures of this section is provided in Appendix A.

The IMPACT heuristic

For determining the customer for insertion, the heuristic calculates *Impact*(u) for each non-routed customer u using Equations (19)–(28), and selects the one that minimises the value of the metric (28). The selected customer u is placed into the route between the ordered pair of customers (i, j) of the route under construction such that LD_u from Equation (25) is minimised. The procedure is repeated until no further non-routed customer can be inserted into the route under construction. In this case, a new route is initialised with a different 'seed' customer and the loop is performed again until all customers are assigned to routes. Then the algorithm terminates by providing the number of routes, the number of active vehicles (equal to the number of routes), the customers that are assigned to each vehicle, the sequence in which customers are visited by vehicles, and the total time–distance–cost of the solution. Note that the feasibility of the solution with respect to constraints (4)–(18) is guaranteed at every step of the algorithm. The new heuristic is presented below:

Algorithm IMPACT

Step 0: Initialisation. Read n, C, c_{ij} (or equivalently t_{ij} or d_{ij}), and e_i, l_i, $\forall\, i, j = 2, \ldots, n$.

Step 1: Select a 'seed' customer to start a route r, finding the furthest customer from the depot. If there is no non-routed feasible customer to start a route, go to Step 6.

Step 2: Find the feasible non-routed customer u that minimises the composite criterion *Impact*(u), ie:

- *Step 2a:* Examine all possible feasible insertions of customer u into the current route. For each feasible insertion, calculate *Impact*(u). Select insertion location that results in minimum *Impact*(u) for this customer.
- *Step 2b:* Repeat Step 2a for all feasible non-routed customers.
- *Step 2c:* Select customer u with minimum *Impact*(u).

Step 3: Insert the selected customer u, to the best insertion location on the current route r (see Steps 2a and 2c). Update the route and set u as a routed customer, decreasing set J.

Step 4: If there are non-routed customers that are feasible for insertion into the current route r, return to Step 2; otherwise proceed to Step 5.

Step 5: If all customers have been scheduled, go to Step 6. Otherwise, go to Step 1 – initiate new route.

Step 6: Terminate; output number of routes (active vehicles), sequence of customers visited by each vehicle, total distance (time) and total cost.

It is important to note that a determinant factor for the effective deployment of IMPACT is the selection of the appropriate parameters embedded in (25) and (28), ie the values of b_1, b_2, b_3, b_s, b_e and b_r. The tuning of these parameters requires experimentation. We proceeded in testing all possible b-weight combinations using a step of 0.1 for each b, on a sample from the classical data sets of Solomon[18] (one problem from each class). The results showed that IMPACT is robust (statistically insignificant differences) with respect to the values of b_1, b_2 and b_3; thus, an equal contribution of these weights was assumed in the remainder of the computational tests. On the other hand, no evident correlation was identified for the weights b_s, b_e and b_r; thus, in our computational tests we examined all combinations of the values for these weights, using a step of 0.2 (ie 0.1, 0.3, ..., 0.9). The only inference we could reach of the test problems was that if time windows were very tight (small), the global disturbance measure of (26) was dominant and the weight b_r, should be set greater than b_s and b_e to achieve good quality solutions.

The heuristic developed above is simple in its mechanics. It fully utilises time windows related information via the criterion *Impact*(u) and employs the typical insertion procedure of Solomon,[18] but expands the search space by preventing infeasibilities due to time windows in a manner similar to Atkinson.[1] Thus, IMPACT differs from previously proposed heuristics for VRPTW in the determination of efficient insertion criteria, its simple implementation and its small computational effort. As proven in the subsequent section, using IMPACT we can solve large problems without high computational requirements, thus making the heuristic practical for real world applications.

Computational results

IMPACT was first tested on the classical data sets R1, C1, RC1, R2, C2 and RC2 of Solomon.[18] Each data set contains problems with 100 customers. The Cartesian coordinates of customers in problem sets R1 and R2 are randomly generated from a uniform distribution. In contrast, sets C1 and C2 have clustered customers. Finally, sets RC1 and RC2 contain semi-clustered customers, ie a combination of clustered and randomly (uniformly) distributed customers. As far as other problem parameters, note that sets R1, C1 and RC1 have tight time windows, short scheduling horizons and vehicle capacity $C = 200$ units, allowing fewer customers per route than the remaining sets. In contrast, problem sets R2, C2, and RC2 have longer scheduling horizons and vehicle capacity C equal to 1000, 700 and 1000 units, respectively, allowing a larger number of customers per route. For additional information concerning the data sets, the reader is referred to the original paper of Solomon.[18]

Table 1(A) shows the average number of routes constructed by IMPACT when applied to Solomon's standard data sets. For comparison purposes, Table 1(A) also includes the results of the following solution approaches when applied to the same data sets: (a) Solomon's I-1 heuristic,[18] (b) the parallel insertion heuristic PARIS,[19] (c) the hybrid exchange (HE) heuristic,[21] (d) the greedy randomised adaptive search procedure GRASP,[22] (e) the tabu search method TABU-A,[23] (f) the reactive tabu search RTS,[24] and (g) the genetic algorithm GENEROUS-20.[25]

Table 1(A) shows that IMPACT improves upon classical heuristic solution approaches, and provides results comparable to those of more computationally expensive meta-heuristics. The fact that IMPACT consistently outperforms GRASP and is close to the other meta-heuristics with respect to solution quality, although the latter approach utilises a random search technique and, thus, examines a larger part of the solution space than simple heuristics (such as IMPACT), reinforces our intuition that exploiting to a large extent time window based information via the metric *Impact*(u) results in high quality solutions. The results of the heuristics concerning the average total route length are shown in Table 1(B), and demonstrate that IMPACT provides relatively good quality solutions with respect to this metric as well (better than traditional heuristics but, as expected, no better than meta-heuristics).

In Table 2, we indicate sets for which our heuristic outperforms (or is equal to) other methods with '*'. The average solutions derived by IMPACT are equal to or better than those of meta-heuristics in more than 50% of the data sets (except RTS). This gives us a further indication that the use of effective measures such as *Impact*(u) can partially overcome the disadvantages of typical construction methods.

In Table 3, we report the results of IMPACT and several meta-heuristic methods on various platforms.[16,24] From Table 3, the computational time efficiency of IMPACT is evident, since IMPACT dramatically reduces the computational time required to derive solutions that are comparable to (or sometimes better than) those of meta-heuristics in the case of problems with randomly distributed customers. It is important to note that, to produce the results of Table 3, we examined all possible b-weight combinations; thus, the computational times are expected to reduce if these weights are fixed.

The results of the new heuristic were also compared to the existing optimal solutions reported by Desrochers *et al.*[14] The computational results are shown in Table 4. For comparison purposes we also provide computational results obtained by other methods. The solutions produced by Solomon's I-1 heuristic,[18] by parallel insertion with Or-opt,[19] by TABU-A,[23] Generous-20,[25] GRASP,[22] and by IMPACT, are shown in the first five columns. The optimal solutions are reported in the last column of Table 4. From Table 4, it is evident that IMPACT achieves the optimal

Table 1 IMPACT *vs* other methods on Solomon's data sets

(A) Average number of routes

Data set	*IMPACT*	*I-1*	*PARIS*	*HE*	*GRASP*	*TABU-A*	*RTS*	*GENEROUS-20*
R1	12.7	13.6	13.3	13.3	12.6	12.6	13.9	12.8
R2	3.1	3.3	3.1	3.3	3.1	3.1	3.1	3.1
C1	10.0	10.0	10.7	10.3	10.0	10.0	10.0	10.0
C2	3.1	3.1	3.4	3.1	3.0	3.0	3.2	3.0
RC1	12.5	13.5	13.4	13.3	12.6	12.6	13.2	12.1
RC2	3.5	3.9	3.6	3.9	3.5	3.4	3.4	3.4

(B) Average total route length

Data set	*IMPACT*	*I-1*	*PARIS*	*HE*	*GRASP*	*TABU-A*	*RTS*	*GENEROUS-20*
R1	1370	1436	1509	1401	1315	1294	1221	1327
R2	1310	1402	1386	1316	1164	1185	1009	1167
C1	865	951	1343	898	827	861	831	846
C2	662	692	797	675	589	602	616	596
RC1	1512	1596	1723	1550	1500	1465	1408	1463
RC2	1483	1682	1651	1635	1414	1476	1303	1402

Table 2 Comparison between literature heuristics and new heuristic on average number of routes

Data set	*I-1*	*PARIS*	*HE*	*GRASP*	*TABU-A*	*RTS*	*GENEROUS-20*
R1	*	*	*	—	—	*	*
R2	*	*	*	*	*	*	*
C1	*	*	*	*	*	*	*
C2	*	*	*	—	—	*	—
RC1	*	*	*	*	*	*	—
RC2	*	*	*	*	—	—	—

Table 3 Average number of routes and CPU time for IMPACT and meta-heuristics

	IMPACT (Pentium 133)		*GE-20 (Sun Sparc 10)*		*TABU-A (Sparc 10)*		*RTS (IBM 6000 RISC)*		*RTS (SGL*100 MHz)*		*TS-CR (Pentium 166)*	
Data set	*A*	*B*	*A*	*B*	*A*	*B*	*A*	*B*	*A*	*B*	*A*	*B*
R1	12.7	141	12.8	264	12.6	639	13.9	586	12.8	450	12.1	5394
R2	3.1	384	3.1	895	3.1	722	3.1	989	3.1	1600	2.7	6115
C1	10.0	168	10.0	248	10.0	435	10.0	273	10.0	540	10.0	644
C2	3.1	294	3.0	952	3.0	431	3.2	409	3.0	1200	3.0	1441
RC1	12.5	126	12.8	270	12.6	586	13.2	564	12.7	430	11.8	2842
RC2	3.5	300	3.4	792	3.4	662	3.4	750	3.6	1300	3.2	3865

A: Mean number of routes.
B: CPU time in seconds.
*: Silicon Graphics Workstation.

Table 4 Comparison between heuristics and existing optimal solutions

Set	*I1*	*PARIS*	*TABU-A*	*GRASP*	*GENEROUS-20*	*IMPACT*	*Optimal*
R101	21	19	19	19	19	19	18
R102	19	18	18	17	17	17	17
C101	10	10	10	10	10	10	10
C102	10	10	10	10	10	10	10
C106	10	11	10	10	10	10	10
C107	10	11	10	10	10	10	10
C108	10	11	10	10	10	10	10

solution with respect to the number of vehicles in six of the seven test problems.

In Table 5, we compare the average optimal solution for the 25, 50 and 100 customer problems reported by Desrochers *et al*[14] to the results obtained by IMPACT. In the first group, the optimal solution is reported for 29 instances, in the second for 14, and in the third for 7 instances.

For the large-scale problems, complex methods such as meta-heuristics spend a lot of computational time to achieve good quality solutions, due to their intricate algorithmic approach. In the literature few heuristics and meta-heuristics are tested on large routing problems with time windows. Two efforts are worth mentioning, each consisting of two problems containing 249 and 417 customers, respectively. The 249 customer problems are based on national and regional fast food delivery applications, and Baker and Schaffer[26] report their solutions. The 417 customer problems are extracted from a fast food routing application in the south-eastern United States.[16] Chiang and Russell[16] solve the two large-scale test problems of 417 customers using Euclidean distances for simplicity. The reactive tabu search achieved solutions of 55 routes for both problem instances. As the authors reported, the problem size allows only one pass (instead of six that is usual). R-tabu found solutions after 38 and 36 minutes of CPU time on a Pentium 166 PC.

Table 5 Comparison between IMPACT and optimal solutions

Problem	*Optimal solution*	*IMPACT*	*Gap*
25 customers (29 problems)	3.8	3.8	0.0%
50 customers (14 problems)	6.9	7.0	1.45%
100 customers (7 problems)	12.1	12.4	2.45%

The above-mentioned problems are still of medium size. Thus, we tested IMPACT on large data sets obtained from the Web.[27] These data sets maintain the features of the

Table 6 Characteristics of large scale problems

Data set	*Total cargo*	*Average earliest time*	*Average latest time*	*Average width TW*	*Time horizon*	*Service time*	*Percentage of time windows*
R1_2_1 200 customers	3513	255.7	265.7	10	634	10	100%
R1_2_2 200 customers	3513	193.2	343.3	150.1	634	10	25%, 50 ⇒ (0,570.3) 75%, 150 ⇒ (257.5,267.5)
R1_2_3 200 customers	3513	118.0	408.0	290.0	634	10	50%, 100 ⇒ (0,570) 50%, 100 ⇒ (236.1,246.1)
R1_4_1 400 customers	7109	311.8	321.8	10.0	804	10	100%
R1_4_2 400 customers	7109	231.7	419.8	188.1	804	10	25%, 100 ⇒ (0,722) 75%, 300 ⇒ (308.9,318.9)
R1_4_3 400 customers	7109	153.0	518.8	365.8	804	10	50%, 200 ⇒ (0,720.8) 50%, 200 ⇒ (306.8,316.8)
R1_6_1 600 customers	10 737	483.7	493.7	10.0	1206	10	100%
R1_6_2 600 customers	10 737	371.3	651.3	280.0	1206	10	25%, 150 ⇒ (0,1087.5) 75%, 450 ⇒ (495.9,505.9)
R1_6_3 600 customers	10 737	244.3	792.5	548.3	1206	10	50%, 300 ⇒ (0,1086.5) 50%, 300 ⇒ (488.5,498.5)
R1_8_1 800 customers	14 230	639.0	649.0	10.0	1688	10	100%
R1_8_2 800 customers	14 230	486.0	874.0	387.5	1688	10	25%, 200 ⇒ (0,1520) 75%, 600 ⇒ (648.6,658.6)
R1_8_3 800 customers	14 230	324.0	1090.6	766.6	1688	10	50%, 400 ⇒ (0,1523.2) 50%, 400 ⇒ (648,658)
R110_1 1000 customers	18 118	788.8	798.8	10.0	1925	10	100%
R110_2 1000 customers	18 118	585.2	1022.0	436.9	1925	10	25%, 250 ⇒ (0,1717.5) 75%, 750 ⇒ (780.2,790.2)
R110_3 1000 customers	18 118	388.5	1254.5	866.0	1925	10	50%, 500 ⇒ (0,1721.8) 50%, 500 ⇒ (777,787)

original sets of Solomon;[18] however, each set contains 200, 400, 600, 800, and 1000 customers, ie the set of customers has a much larger cardinality than the customer data sets of Solomon.[18] In each problem, the objective was to minimise the number of routes. IMPACT was applied on these large-scale problems, the characteristics of which are shown in Table 6.

In order to reduce the computational time we relaxed some measures of the criteria of the two phases of the original heuristic. For evaluation reasons we executed the original code for the insertion heuristic reported by Baker and Schaffer.[26] Table 7 shows the solutions obtained by IMPACT and the heuristic of Baker and Schaffer (B&S). The computational time increases when the percentage of time windows decreases in the data set. This happens because the number of feasible customers for insertion increases. The good quality solutions produced in reasonable computational time render IMPACT useful for real world applications. In conclusion, IMPACT derives the best results for problems where customers are randomly distributed and time windows are tight. This result holds whether the data sets contain small or large number of customers. This is not surprising, because the *Impact*(u) criterion for customer selection for insertion causes the minimum possible impact on routed and non-routed customers' time windows.

Table 8 shows the results of IMPACT for the 600 customers randomly generated set R1 and for the 600 customers mixed randomly and clustered generated set

Table 7 Comparison of B&S and IMPACT. Mean number of routes and CPU times on a 450 MHz PC

Data set	*Baker and Schaffer*	*IMPACT*	*CPU time (min)*
R1-2-1 (200 customers)	24	21	0.18
R1-2-2 (200 customers)	22	18	0.21
R1-2-3 (200 customers)	23	18	0.26
R1-4-1 (400 customers)	50	41	0.9
R1-4-2 (400 customers)	43	36	1.3
R1-4-3 (400 customers)	42	36	1.8
R1-6-1 (600 customers)	72	62	2.8
R1-6-2 (600 customers)	70	55	4.7
R1-6-3 (600 customers)	64	54	6.0
R1-8-1 (800 customers)	104	81	7.0
R1-8-2 (800 customers)	91	72	14.8
R1-8-3 (800 customers)	72	72	14.7
R110-1 (1000 customers)	128	101	13.5
R110-2 (1000 customers)	121	92	22.2
R110-3 (1000 customers)	107	91	29.1

Table 8 Solution quality *vs* CPU time for IMPACT on 600-customer problems

Data set	*Lower bound on route no.*	*IMPACT (route no.)*	*CPU time (Pentium 450)*
R1 (10 problems)	53.7	55.2	5.0
RC1 (10 problems)	54.4	55.5	4.6
R2 (10 problems)	10.8	11.4	8.5
RC2 (10 problems)	10.9	13.5	6.9

RC1. The second column in Table 8 is the lower bound on the number of routes, which is given by the ratio of the total customer demand and the vehicle capacity $[= (\sum q_i)/C]$. The third column is the solution obtained by IMPACT and the last column is the computational time on a Pentium 450 MHz PC. For the randomly generated set R1 and R2 our approach gave slightly better results comparing with the lower bound.

Finally, in Table 9 we present the percentage difference between the number of vehicles provided by our heuristic and the lower bound of this number. The latter is calculated as follows:

$$\text{trend} = (\text{lower bound on route no.} - \text{IMPACT solution})/\text{lower bound on route no.} \times 100 \quad (29)$$

As we can see the percentage inclination to the lower bound of the number of routes decreased as the number of customers increased.

Industrial application

The industry-based problems were obtained from a delivery company, which supplies goods to a large number of supermarkets throughout the Athens Metropolitan area, in Athens, Greece.[28] The number of customers was approximately two thousand (1943 to be exact). A map of the Athens Metropolitan area via the MapInfo Geographical Information System (GIS) used by the company provides the coordinates of the customers. In Figure 3, we show the actual geographical distribution of customers.

For the particular instance of the problem we solve, the company uses a fleet of 72 vehicles to perform all daily deliveries to customers. The company owns some of these vehicles, while other vehicles are used on a case-by-case basis; the latter are private trucks owned by transportation companies and operated by certified drivers.

The time window associated with the depot is 3300 time units, which is equivalent to one shift. As a result, many customers can be serviced on the same route by a single vehicle during one shift. The vehicle capacity is 1000 units. A fixed service time for vehicle unloading is associated with each customer. This service time is 10 time units per loading or unloading operation. Without loss of generality, and to be consistent with the examples reported in the literature, we assumed that travel times are equivalent to the corresponding Euclidean distances. All customers pose constraints on the time of service (time windows). The time windows' length is fixed at 150 time units. The earliest service time for each customer is set during the issue of a sales order. This time, as well as the latest service time determined by the time window's length, is considered a hard constraint for the vehicle routing problem. This is because both these times cannot be violated by the vehicles and, thus, deliveries to a specific customer cannot be performed before the earliest service time or after the latest service time associated with this customer.

We formulated the vehicle routing problem with time windows associated with this instance of the industrial situation. We solved the resulting problem using algorithm IMPACT. To accelerate the generation of a near-optimum solution, we used the following parameters weights for the algorithm: $b_1 = b_2 = b_3 = 1/3$ for the measure of local disturbance LD_u; $b_s = 0.1, b_e = 0.2$, and $b_r = 0.7$ for the own, external and internal impact metrics, respectively.

With the above parameters IMPACT provided the best solution of 40 routes after less than a minute on a Pentium II 450 MHz PC. This is a dramatic improvement over the fleet size currently operated by the company (72 vehicles) for this problem instance, even after factoring in the actual distances (compared to the approximation by Euclidean distances). Clustering the set of customers into four parts (in Figure 3 the four clusters are obvious) and running the code for each cluster produced the total solution of 39 routes. This is an indication of the robustness of algorithm IMPACT, and a demonstration of the algorithm's capabilities in providing high quality solutions without resorting to two-phase cluster-first route second approaches.

Conclusions

In this paper we have developed a new heuristic to solve the vehicle routing problem with time windows. Our

Table 9 Trends of new heuristic for large-scale problems

Data set	*Lower bound on route no.*	*Heuristic*	*Gap*	*Trend*
R1-2 (10 problems of 200 customers)	17.6	18.4	0.8	4.5%
R1-4 (10 problems of 400 customers)	35.5	36.8	1.3	3.6%
R1-6 (10 problems of 600 customers)	53.7	55.2	1.5	2.8%
R1-8 (3 problems of 800 customers)	71.1	73.0	1.9	2.6%
R1-10 (3 problems of 1000 customers)	90.6	92.8	2.2	2.4%

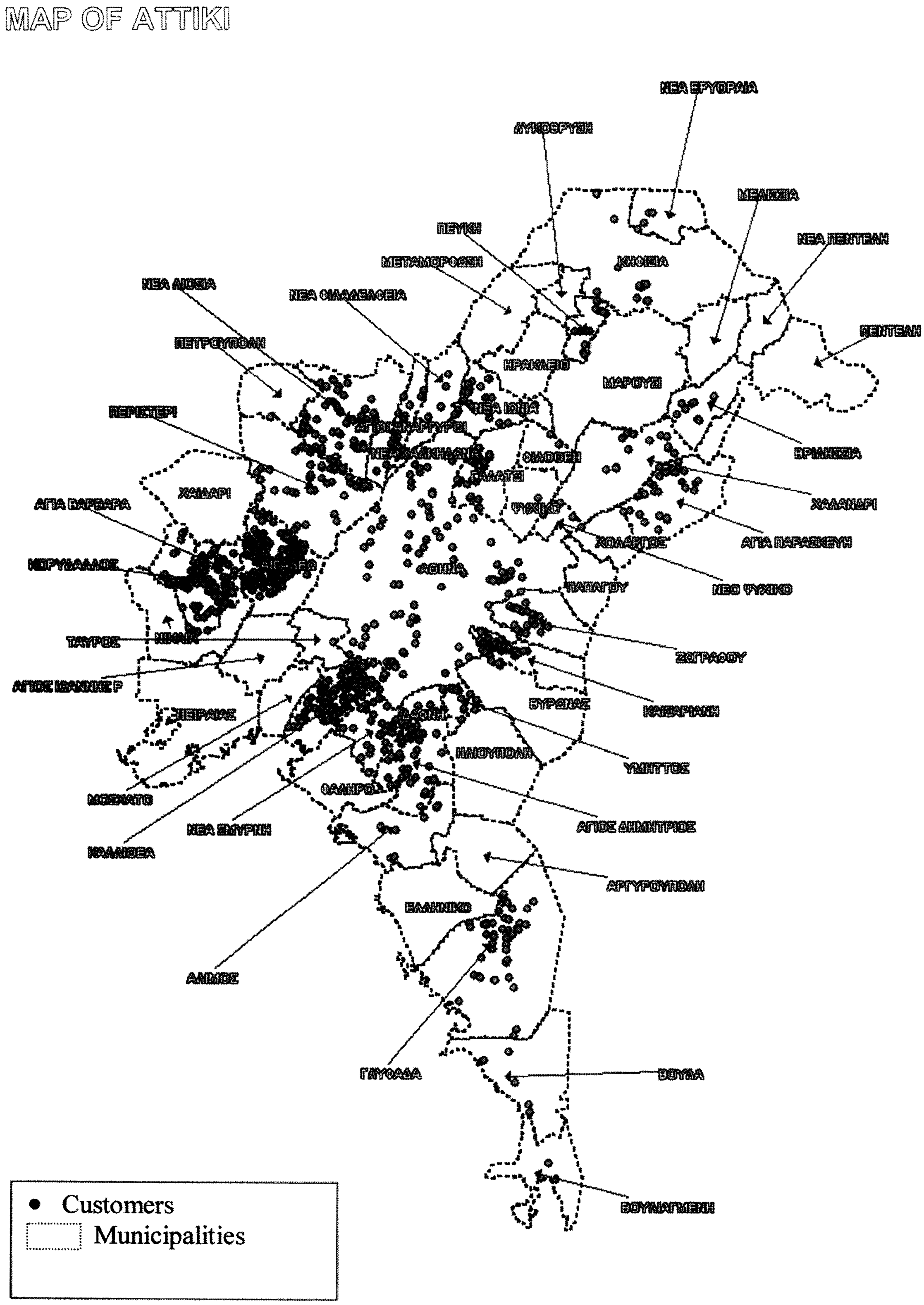

Figure 3 The distribution of customers of the industrial problem.

approach, which is based upon Atkinson's[1] greedy look-ahead framework for combinatorial optimisation problems, utilises time windows related information via composite customer selection and route insertion criteria. These criteria exploit the interrelationships between customers introduced by time windows that dictate the sequence in which vehicles must visit customers. The algorithm IMPACT performs very well on test problems from the literature, providing high quality solutions with respect to the number of vehicles required, within short computational times. IMPACT is simple and easy to implement and to apply to vehicle routing problems with minimal computational effort. The results presented indicate that the heuristic provides solutions that are competitive with the best solutions of meta-heuristics for a large number of literature problems. Comparisons with optimum solutions show that IMPACT reaches near optimal solutions for most problem instances. Our results provide further validation for the general problem solving methodology proposed by Atkinson.[1]

We have applied our solution method in a real example from the food industry. The results reveal that our approach is suitable for industrial applications since it provides very good quality solutions to large examples in very short computational times. The latter is a prerequisite for deployment of vehicle routing solution methods in actual industrial environments, where users require solutions fast in order to be able to adjust schedules, number of drivers–vehicles, customers to be visited and time windows, if necessary. The real world is dynamic; thus effective and efficient decision support tools are needed to help practitioners run their daily operations in the best possible manner. We believe that IMPACT is a step towards this direction and we are currently examining the possibility of directly interfacing it to the MapInfo GIS platform for delivery and other routing applications that are used in industrial environments.

In terms of future research directions, it is important to consider the problems of vehicle routing with soft time windows and the applicability of methods such as IMPACT to this version of VRPTW. Furthermore, the development of coherent vehicle routeing solution within Geographical Information Systems is a research direction worth pursuing, especially with respect to the dynamic and repetitive application of simple heuristic methods that will overcome the disadvantages of mathematical programming methods and meta-heuristic implementations.

Appendix A: an illustrative example

To illustrate the metrics/criteria used within IMPACT, we consider problem R101 of Solomon; for the problem parameters the reader can refer to the original work of Solomon.[18] We assume that we have arrived at a partial solution when the route under construction comprises of the sequence of customers $\{1\text{-}48\text{-}9\text{-}47\text{-}18\text{-}1\}$, as in Figure A-1.

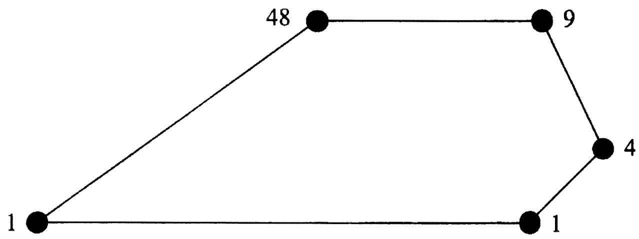

Figure A-1 Route under construction for example R101.

We are interested in estimating the disturbance of non-routed customers 8 and 94, which are the only customers that can be inserted in route $\{1\text{-}48\text{-}9\text{-}47\text{-}18\text{-}1\}$, between customers (48, 9), and (18, 1), respectively. The time windows of these customers are shown in Figure A-2.

Given that $d(48, 8) = 13.4$, $d(8, 9) = 12.2$, and $d(48, 9) = 13.1$, the route distance increment for the insertion of customer 8 in the current route, according to (22), is

$$c_{1,8}(48, 9) = 13.4 + 12.2 - 13.1 \Leftrightarrow c_{1,8}(48, 9) = 12.5 \tag{A1}$$

Given that for customer 9 the latest service time is 105, the insertion of customer 8 in the route causes urgency on customer 9 that according to (23) is

$$\begin{aligned} c_{2,8}(48, 9) &= [l_9 - (a_{48} + s_{48} + d_{48,9})] \\ &\quad - [l_9 - (a_8 + s_8 + d_{8,9})] \Leftrightarrow \\ c_{2,8}(48, 9) &= [105 - (51 + 10 + 13.1] \\ &\quad - [105 - (81 + 10 + 12.2)] \Leftrightarrow \\ c_{2,8}(48, 9) &= 30.9 - 1.8 = 29 \end{aligned} \tag{A2}$$

In (A2), $a_{48} = max\{d_{1,48}, e_{48}\} = 51$, and $a_8 = max\{a_{48} + s_{48} + d_{48,8}, e_8\} = 81$, since 48 is the first customer visited by the vehicle after the depot (which is denoted by index 1).

Using Equation (24) we can calculate $c_{3,u}$:

$$\begin{aligned} & c_{3,8}(48, 9) = l_8 - (a_{48} + s_{48} + d_{48,8}) \Leftrightarrow \\ & c_{3,8}(48, 9) = 91 - (51 + 10 + 13.4) \Leftrightarrow c_{3,8}(48, 9) = 16.6 \end{aligned} \tag{A3}$$

From Equation (25), we can substitute the results of (A1)–(A3) and calculate the local disturbance that the insertion of customer 8 between customers 48 and 9 causes on the current route:

$$\begin{aligned} & LD_8(48, 9) = b_1 c_{1,8} + b_2 c_{2,8} + b_3 c_{3,8} \Leftrightarrow \\ & LD_8(48, 9) = (1/3) \times 12.5 + (1/3) \\ & \qquad \times 29.1 + (1/3) \times 16.6 \Leftrightarrow LD_8(48, 9) = 19.2 \end{aligned} \tag{A4}$$

In (A4) we assumed a balanced scheme for the contribution of the c's, ie, $b_1 = b_2 = b_3 = 1/3$.

Since we have a single possible insertion point in the route, Equation (26) is redundant, and the global disturbance is

$$IR_8 = LD_8(48, 9) \Leftrightarrow IR_8 = 19.2 \qquad \text{(A5)}$$

For customer 8 the accessibility is the reverse of the global disturbance; thus, substituting (A5) in Equation (27), we get

$$ACC_8 = (19.2)^{-1} \Leftrightarrow ACC_8 = 0.052 \qquad \text{(A6)}$$

Using Equation (19) we can calculate the own impact, ie the coverage of the selected customer's time window that results from his insertion into the current route:

$$IS_8 = a_8 - e_8 \Leftrightarrow IS_8 = 81 - 81 \Leftrightarrow IS_8 = 0 \qquad \text{(A7)}$$

Using Equation (21) we can calculate the external impact of the customer over all time windows of current non-routed customers 8:

$$IU_8 = 44.2 \qquad \text{(A8)}$$

Using Equation (28) we can substitute the values of the three different impacts and calculate the $Impact(8)$ of the insertion of customer 8 between customer 48 and 9:

$$\begin{aligned} &Impact(8) = b_s IS_8 + b_e IU_8 + b_r IR_8 \Leftrightarrow \\ &Impact(8) = (1/3) \times 0 + (1/3) \times 44.2 + (1/3) \\ &\qquad \times 19.2 \Leftrightarrow Impact(8) = 21.1 \end{aligned} \qquad \text{(A9)}$$

Similarly, in (A9) we assume a balanced scheme, ie $b_s = b_e = b_r = 1/3$.

Given that $d(18, 94) = 14.3, d(94, 1) = 20.2$ and $d(1, 18) = 30.4$, the insertion of customer 94 causes a distance increase according to (22):

$$\begin{aligned} c_{1,94}(18, 1) &= 14.31 + 20.24 - 30.41 \Leftrightarrow c_{1,94}(18, 1) \\ &= 4.1 \end{aligned} \qquad \text{(A10)}$$

The insertion of customer 94 causes urgency on depot 1; using Equation (23) we get

$$\begin{aligned} c_{2,94}(18, 1) &= (230 - (157 + 10 + 30.41)) \\ &\quad - (230 - (188 + 10 + 20.2)) \Leftrightarrow \\ c_{2,94}(18, 1) &= 32.6 - 11.8 \Leftrightarrow c_{2,94}(18, 1) \\ &= 20.8 \end{aligned} \qquad \text{(A11)}$$

The last metric of the internal impacts gives

$$\begin{aligned} c_{3,99}(18, 1) &= 198 - (157 + 10 + 14.3) \Leftrightarrow c_{3,99}(18, 1) \\ &= 16.7 \end{aligned} \qquad \text{(A12)}$$

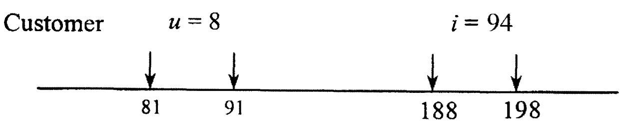

Figure A-2 Time windows of customers 8 and 94 of the example.

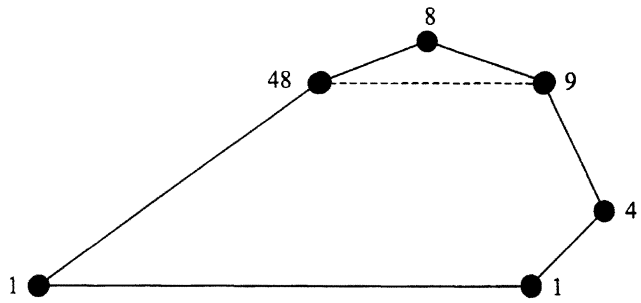

Figure A-3 New route after insertion of customer 8.

Following Equation (25) the local disturbance is:

$$\begin{aligned} LD_{94}(18, 1) &= (1/3) \times 4.1 + (1/3) \times 20.8 + (1/3) \\ &\quad \times 16.7 \Leftrightarrow LD_{94}(18, 1) = 13.8 \end{aligned} \qquad \text{(A13)}$$

and

$$IR_{94} = 13.8 \qquad \text{(A14)}$$

The accessibility of customer 94 is

$$ACC_{94} = (IR_{94})^{-1} = 0.072 \qquad \text{(A15)}$$

According to (19):

$$IS_{94} = a_{94} - e_{94} \Leftrightarrow IS_{94} = 188 - 188 \Leftrightarrow IS_{94} = 0 \qquad \text{(A16)}$$

According to (21):

$$IU_{94} = 57.6 \qquad \text{(A17)}$$

According to Equation (28):

$$\begin{aligned} &Impact(94) = (1/3) \times 0 + (1/3) \times 13.8 + (1/3) \times 57.6 \Leftrightarrow \\ &Impact(94) = 23.1 \end{aligned} \qquad \text{(A18)}$$

Customer 8 has better total impact than customer 94 does, and it is selected for insertion into the current route (Figure A-3). It is worth noting, however, that although customer 94 has better accessibility (internal impact) than customer, 8, it is not selected for insertion due to the summation of all possible impacts.

Acknowledgements—The authors would like to thank the two anonymous referees and the editor, Dr J Ranyard, for their acute comments and useful suggestions that helped improve the content and the presentation of the paper.

References

1 Atkinson JB (1994). A greedy look-ahead heuristic for combinatorial optimisation: an application to vehicle scheduling with time windows. *J Opl Res Soc* **45**: 673–684.
2 Desrosiers J, Dumas Y, Solomon MM and Soumis F (1995). Time constrained routing and scheduling. In: Ball M, Magnanti TL, Monma CL and Nemhouser GL (eds). *Network Routing, Handbook in Operations Research and Management Science 8*. Elsevier: Amsterdam, pp 35–139.
3 Golden B and Assad AA (1988). *Vehicle Routing: Methods and Studies*. Elsevier Science Publishers: Amsterdam.

4 Ahn BH and Shin JY (1991). Vehicle-routeing with time windows and time-varying congestion. *J Opl Res Soc* **42**: 393–400.
5 Gendreau M, Laport G and Potvin J (1997). Vehicle routing: modern heuristics. In: Aarts E and Lenstra JK (eds). *Local Search in Combinatorial Optimisation*, John Wiley & Sons: New York, pp 311–336.
6 Bodin L, Golden B, Assad AA and Ball M (1983). Routing and scheduling of vehicles and crews: the state of the art. *Comput Opers Res* **10**: 62–212.
7 Desroshers J, Lenstra JK, Savelsbergh MWP and Soumis F (1988). Vehicle routing with time windows: optimization and approximation. In: Golden BL and Assad AA (eds). *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers: Amsterdam, pp 65–84.
8 Lenstra KJ and Rinnooy Kan AHG (1981). Complexity of vehicle routeing and scheduling problems. *Networks* **11**: 221–227.
9 Laporte G (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *Eur J Opl Res* **59**: 345–358.
10 Atkinson JB (1990). A vehicle-scheduling system for delivering school meals. *J Opl Res Soc* **41**: 703–711.
11 Atkinson JB (1998). A greedy randomised search heuristic for time-constrained vehicle scheduling and the incorporation of a learning strategy. *J Opl Res Soc* **49**: 700–708.
12 Kohl N and Madsen OBG (1997). An optimisation algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Opns Res* **45**: 395–406.
13 Fisher M, Jörnsten KL and Madsen OBG (1997). Vehicle routing with time windows: two optimisation algorithms. *Opns Res* **45**: 488–492.
14 Desrochers M, Desrosiers J and Solomon MM (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Opns Res* **40**: 342–354.
15 Kolen AWJ, Rinnooy Kan AHG and Trienekens HWJM (1987). Vehicle routing with time windows. *Opns Res* **35**: 266–273.
16 Chiang W-C and Russell R (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS J Comp* **9**: 417–430.
17 Desrosiers J, Lenstra JK and Savelsbergh MWP (1990). A classification scheme for vehicle routing and scheduling problems. *Eur J Opl Res* **46**: 322–332.
18 Solomon MM (1987). Algorithms for the vehicle routing and scheduling problems with time windows constraints. *Opns Res* **35**: 254–265.
19 Potvin JY and Rousseau JM (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur J Opl Res* **66**: 331–340.
20 Savelsbergh, MWP (1986). Local search for routing problems with time windows. *Ann Opns Res* **4**: 285–305.
21 Potvin JY and Rousseau J-M (1995). An exchange heuristic for routing problems with time windows. *J Opl Res Soc* **46**: 1433–1446.
22 Kontoravdis G and Bard J (1995). A GRASP for the vehicle routing problem with time windows. *INFORMS J Comp* **7**: 10–23.
23 Potvin JY, Kervahut T, Garcia BL and Rousseau JM (1996). The vehicle routing problem with time windows part I: tabu search. *INFORMS J Comp* **8**: 158–164.
24 Taillard E *et al* (1997). A tabu search heuristic for the vehicle routing problem with soft time windows, *Trans Sci* **31**: 170–186.
25 Potvin JY and Bengio S (1996). The vehicle routing problem with time windows Part II: genetic search. *INFORMS J Comp* **8**: 165–172.
26 Baker E and Schaffer JR (1986). Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *Am J Math Mngt Sci* **6**: 261–300.
27 Homberger J. Extended Solomon's VRPTW instances. http://www.fernuni0hagen.de/WINF/touren/inhalte/probinst.htm
28 Ioannou G, Kritikos M and Prastacos GP (1999). Vehicle routing applications in the food and beverage industry of Greece. Technical Report (in Greek), Graduate Program in Decision Sciences, Athens University of Economics and Business, Athens, Greece.

Received November 1999;
accepted November 2000 after two revisions