The Real-Time Vehicle Routing Problem

Conference Paper · January 2007

DOI: 10.1007/978-3-540-77903-2_22 · Source: DBLP

Knut Richter
Saint Petersburg State University

Vehicle routing problem with real-time travel times

Irena Okhrin*

Department of Information & Operations Management,
European University Viadrina,
P.O. Box 1786, 15207 Frankfurt (Oder), Germany
Fax: +49 335 5534 2675 Email: irena.okhrin@euv-frankfurt-o.de
*Corresponding author

Knut Richter

Department of Industrial Management,
European University Viadrina,
P.O. Box 1786, 15207 Frankfurt (Oder), Germany
Email: richter@euv-frankfurt-o.de

Abstract: The paper considers a vehicle routing problem with time windows and real-time travel times. We assume the deployment of an information and communication system that is based on mobile technologies, which provides a real-time mobile connection between the dispatching centre and drivers, allows localising vehicles on road, and gives the online overview over traffic conditions. We explicitly incorporate the possibility to react to some dynamic events like traffic impediments and divert a vehicle en route away from its current destination. We formulated the vehicle routing problem with real-time travel times as a mixed-integer linear programming model and developed a genetic algorithm to solve it. Moreover, we performed an extensive computational study to prove the efficiency of the proposed algorithm on well-known static benchmarks and to test its performance in dynamic settings.

Keywords: vehicle routing problem; time-dependent; real-time travel times; mobile information and communication technology; genetic algorithm.

Reference to this paper should be made as follows: Okhrin, I. and Richter, K. (2009) ‘Vehicle routing problem with real-time travel times’, *Int. J. Vehicle Information and Communication Systems*, Vol. 2, Nos. 1/2, pp.59–77.

Biographical notes: Irena Okhrin is an Assistant Professor in Information & Operations Management at the European University Viadrina in Frankfurt (Oder). She has received a Diploma in Applied Mathematics and a Master in Economics degree from Ivan Franko National University of Lviv, Ukraine and a doctor’s degree in Economics from the European University Viadrina. Her activities are devoted to the analysis of the influence of wireless technology on logistics business processes and supply chain management. Her research interests also include mathematical modelling and new information technology.

Knut Richter is currently a Professor in the Department of Industrial Management at the European University Viadrina in Frankfurt (Oder). He has teaching experience as Guest and Associate Professor at universities in numerous countries like Germany, Ethiopia, Russia, Hungary and Uruguay, among others. He has been a Coordinator of seven projects on mathematical modelling and management at the enterprise level. His research and teaching are in the areas of logistics and operations management, mathematical optimisation, operations research and environmental management. His works have been published in various academic journals, including the *International Journal of Production Economics*, *European Journal of Operational Research*, *Central European Journal of Operations Research*, *OR Spectrum* and others.

1 Introduction

Vehicle Routing Problems (VRPs) appear in distributing and/or collecting of goods by commercial or public transport. The aim of a VRP is to determine a route and schedule of vehicles in order to satisfy customer orders and minimise operational costs. The objective is usually to minimise travel times, distances, costs or the number of used vehicles while distributing goods. In addition, many side constrains have to be satisfied, such as vehicle capacity constrains, route duration constrains, time window constrains, etc.

In the past, vehicles executing routes and dispatchers in the control centre were acting separately, without or with only little information exchange. The position of vehicles en route was not known to the dispatcher and it was not always possible to establish a good connection with drivers. Therefore, all client requests that arrive after vehicles departure have to be postponed till the next planning period. Recent advances in information and communication technologies improve dramatically the quality of communication between drivers and the dispatching office. New customer orders as well as route changes can now be easily communicated to drivers, thus enhancing service quality and reducing costs. Moreover, state-of-the-art navigation systems provide real-time traffic and weather conditions allowing to escape hampered roads.

In the paper, we consider a dynamic vehicle routing problem with time windows and real-time travel times. To enhance the effectiveness of operations management, we assume the deployment of an information and communication system that is based on mobile technologies. It encompasses, first of all, a real-time mobile connection between the dispatching centre and the drivers. Second, the dispatching office can localise the vehicles on road via the Global Positioning System (GPS) facilitating fleet utilisation and increasing productivity. And third, modern technologies are able to provide the overview over current traffic conditions, what enables the consideration of time-dependent travel times that are permanently updated during the day. All mentioned possibilities of the mobile technology allow reacting to some dynamic events like traffic impediments or new customer request for service and divert a vehicle on road in real time. We formulate the VRP with online travel time information as a mixed-integer linear programming model. To solve the problem we develop a genetic algorithm and test its performance on well-known benchmarks. The achieved results are competitive with best published solutions.

The contributions of the paper are threefold. First, we discuss the deployment of modern information and communication systems based on wireless technology in vehicle routing. Such systems are an indispensable source of real-time data on fleet location and traffic conditions and serve as a communication medium between vehicles and dispatchers. They allow using estimations of real travel times in online mode, as provided by navigation systems, thus eliminating the need to refer to some constant travel times or base calculations on a step function that is known ex ante. Second, we define a framework for a VRP with real-time travel times and formulated a mathematical model which allows to consider variable travel times and to capture the dynamics of real-life systems. By introducing some intermediate artificial point in the road network we explicitly consider the possibility to divert a vehicle to other destination, if the planned route is hampered. Thus, vehicles arrive always on time, all customers are served within provided time windows and costs are minimised. Finally, we developed an algorithm that can deal with the complexity of the proposed model and test its performance for constant and variable travel times.

The rest of the paper is organised as follows. Section 2 outlines the application of mobile technologies to vehicle routing. Section 3 presents the literature review. Further, the time-dependent VRP with real-time travel times is described in Section 4 and in Section 5 a corresponding mathematical model is formulated. The genetic algorithm to solve the developed model is elaborated in Section 6. Finally, Section 7 presents the computational results for both constant and variable travel times and Section 8 concludes.

2 Mobile technologies in vehicle routing

The novelty of mobile technology is in its ability to provide accurate and critical information in a timely manner. For systems that are based on dynamic vehicle routing the flow of information is as important as the flow of physical assets (Cetin and List, 2006). Therefore, their practical implementation is not possible without a real-time, high-bandwidth information and communication system. Recent developments in wireless communication technologies make the exchange of information between the dispatching centre and the vehicles on-road as well as inter-vehicle data exchange especially cost-effective and uncomplicated. Thus, new instructions for action or recent customer orders can be sent to drivers at any time, regardless of their location and status. In addition, the wireless technology helps to capture up-to-the-minute traffic data and sends it to a powerful workstation. The back-end software analyses the data and transforms it into the useful information about current traffic conditions and estimations of travel times. This information is in real-time transmitted to drivers through in-vehicle route guidance devices and Advanced Traveller Information System (ATIS) facilitating more informed decision making.

Furthermore, based on signals from satellites, the GPS is used to determine current vehicle positions, dramatically improving fleet monitoring and management. GPS, together with Geographic Information Systems (GIS), is a power tool for data integration, visualisation and analysis. Integrated GPS and GIS are a valuable source of information for the dispatching centre that can locate the vehicles on road and display their position on an electronic road map. In addition, dispatchers can determine the distance between various vehicles of the fleet as well as between a vehicle and its final destination (Küpper, 2005). This improves fleet utilisation, decreases reaction time, and gives the overview over routes execution.

3 Literature review

The vehicle routing problem was firstly introduced by Dantzig and Ramser (1959) in their seminal paper about the delivery of gasoline. Since that time a vast body of research was devoted to a generic problem and its numerous extensions. Usually the problem deals with distribution or collection of goods. However, it is also relevant for delivery of petroleum and gases, courier services, rescue and repair services, emergency services, taxi cab and tramp ship operations, combined pick-up and delivery services as well as for the transportation of elderly and disabled people (Psaraftis, 1995; Ghiani et al., 2003). A variety of exact and approximate solution methods were developed for VRP. For the survey of some of them see, for example, Laporte (1992); Golden et al. (1998); Laporte et al. (2000), etc. Besides this, the two books edited by Golden and Assad (1988) and Toth and Vigo (2002) were published on the topic.

In practice, vehicle routing problems are mostly dynamic; in literature, however, the static version of a problem is handled more often. The dynamic or real-time version of a vehicle routing problem was investigated by Psaraftis (1988, 1995). Earlier works on the problem also belong to Bertsimas and van Ryzin (1991, 1993). Gendreau and Potvin (1998) review several variations of a dynamic VRP such as dial-a-ride problems, repair and courier services as well as express mail delivery. A dynamic pickup and delivery problem was studied by Swihart and Papastavrou (1999) and recently by Mitrović-Minić et al. (2004). Attanasio et al. (2004) examined a dynamic dial-a-ride problem. The invited review on the dynamic vehicle routing was presented by Ghiani et al. (2003). Some recent works on the topic include, among others, Taniguchi and Shimamoto (2004); Haghani and Jung (2005); Potvin et al. (2006), etc.

The majority of papers on vehicle routing make an assumption about constant travel times that are a scalar transformation of distances. This concept, however, does not represent well ever changing travel times of the real world. The notion of time-dependent travel times for vehicle routing was introduced by Malandraki (1989) and further investigated by Malandraki and Daskin (1992) and Malandraki and Dial (1996). The authors assumed that travel times between two nodes change throughout the day in an a priori known fashion. This partially captures the predictive variations of the travel time function. They divide a day into a few time intervals and define travel time variation as a step function. This function, however, does not satisfy the First-In-First-Out (FIFO) assumption. That is, if two vehicles depart from the same depot and travel to the same customer using the same roads, the vehicle that left the depot later may arrive earlier to its destination compared to its rival.

The situation is improved in the subsequent papers. Ichoua et al. (2003) proposed a model for the vehicle routing problem based on the time-dependent travel speed which satisfies the FIFO property. The travel speed changes as time proceeds from one time interval to another. The work by Chen et al. (2006) also defines travel time between nodes as a step function, it is however unknown in advanced and may change if an unexpected event occurs. In addition, Jung and Haghani (2001) and Haghani and Jung (2005) consider a continuous travel time function and develop a genetic algorithm for the corresponding VRP. Finally, Donati et al. (2008) present the multi-ant colony system to simultaneously optimise the number of tours and the total travel time of the time-dependent VRP in urban context.

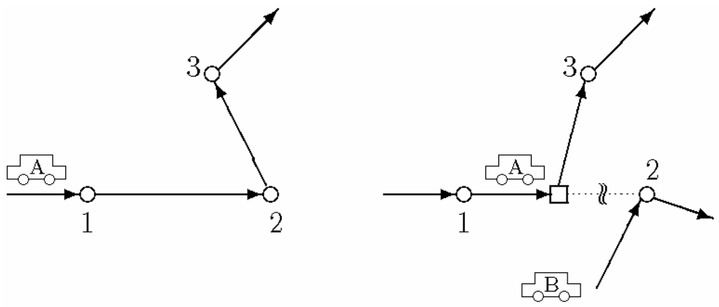
4 Problem definition

We consider a dynamic vehicle routing problem with time windows and online travel time information. The problem is defined on a complete graph $G = (V, A)$, where $V = 0, 1, \ldots, n$ is the vertex set and $A = \{(i,j): i,j \in V, i \neq j\}$ the arc set. Vertex 0 represents a depot whilst other vertices represent geographically dispersed customers that have to be served. A positive deterministic demand is associated with every customer. The demand of a single customer cannot be split and should be serviced by one vehicle only. Each customer defines its desired period of time when he wishes to be served. A set of K identical vehicles with capacity Q is based at the single depot. Each vehicle may perform at most one route which starts and ends at the depot. The vehicle maximum load may not exceed the vehicle capacity Q. The objective is to design routes on G such that every customer belongs to exactly one route and the total travel time of all vehicles is minimised.

Not all customer requests are known at the beginning of the planning period (typically, a day). Some of them are revealed in the course of time, when vehicles are already in transit. This fact makes the problem dynamic or time-dependent. We assume that the policy of the company is to accept all received orders. Another dynamic component of the problem is a link travel time function between a pair of nodes. The frequently considered constant travel time function, proportional to the distance between the nodes and time independent, is not realistic. It does not represent well the real-world situation, where travel times vary with time and are subject to numerous factors. In practice, the link travel time function fluctuates because of changing traffic and weather conditions like, for example, congestion during rush hours, accidents, etc. Furthermore, available models for the dynamic vehicle routing usually imply that a vehicle en route must first reach its current destination and only after that its route can be changed to incorporate a new customer or to escape a hampered road. It is not possible to divert a vehicle in transit between two nodes in response to a new customer request or changing travel times. Exactly on the way to its immediate destination, however, the vehicle may encounter an unpredicted congestion or other traffic impediment. So, the vehicle has to wait unreasonably long, endangering the satisfaction of the time window restrictions of the subsequent clients. Instead of this, it can deviate from its route and serve other customers in the meantime.

Thanks to mobile technology, we can overcome the mentioned shortcomings and model vehicle routing in more realistic settings. We assume that there is a real-time communication between the vehicles and the dispatching centre. Furthermore, the dispatcher has an overview over current traffic conditions and knows the locations of the vehicles. Therefore, we explicitly incorporate into the model varying and time-dependent travel times and the possibility to divert a vehicle en route. To gain an opportunity to direct a vehicle in transit to another customer, we insert an additional node into the original road network at the place of vehicle location. Thus we create the so-called artificial intermediate nodes for these vehicles that at time of routes adjustment are in transit between two nodes. No demands are associated with these nodes and the vehicles must immediately leave them. This concept is illustrated in Figure 1. Let us assume that on its way from node 1 to node 2 vehicle A encounters a hampered road. In order to avoid unreasonable waiting, we create an artificial intermediate node (denoted by a square) and run a re-optimisation procedure to find better vehicle routes under given circumstances. After routes adjustment, vehicle A is diverted to node 3 and node 2 is visited by vehicle B instead. Hence, we do not commit customers to routes. Clients can be served by any vehicle that has enough capacity, what increases the flexibility of the system.

Figure 1 Anticipated and updated vehicle routes



To better react to changing link travel times or to new customer requests, we use the concept of time rolling horizon. We formulate a series of mixed-integer linear programming models, where each model characterises a particular static vehicle routing problem with heterogeneous fleet at a specific point of time. The re-optimisation algorithm is then performed on the graph that besides the initial road network includes also the artificial intermediate nodes. The model is formulated and solved every time τ when one of the following dynamic events happens:

- A new customer request for service has arrived (if service requests arrive too late to be served in the current planning period, they are postponed till the next period and no routes adjustment is undertaken).
- Travel time between a pair of nodes is updated.
- A vehicle is considerably ahead/behind the schedule.

After a new event has arrived, the task of the dispatching centre is to process recent data, determine new routes, redirect vehicles en routes, update schedules, and finally inform customers about anticipated arrival times. The routes adjustment procedure should take into account new travel time information, current vehicle locations and the most recent service requests that arrived after the latest adjustments in routes. The time between two subsequent routes adjustments we will call the execution period. Instead of re-optimising the vehicle routes after the arrival of every dynamic event (what may be impractical), it is also possible to introduce a tolerance parameter that buffers the incoming events before further processing. In this case the dispatcher should wait until several dynamic events occur and only then run the re-routing algorithm. According to Potvin et al. (2006) this strategy is more beneficial if compared with immediate reaction to every single dynamic event.

5 The time-dependent vehicle routing problem

5.1 Notations

Data sets

N_0 depot

N_d $\equiv N_{demand}(\tau)$ set of unserved demand nodes at time τ

N_s $\equiv N_{serviced}(\tau)$ set of nodes being served at τ

N_a $\equiv N_{artificial}(\tau)$ set of artificial intermediate nodes for vehicles being en route at time τ

N_{0d} $\equiv N_0 \bigcup N_d$

N_{ds} $\equiv N_d \bigcup N_s$

N_{0sa} $\equiv N_0 \bigcup N_s \bigcup N_a$

N_{dsa} $\equiv N_d \bigcup N_s \bigcup N_a$

N_{0dsa} $\equiv N_0 \bigcup N_d \bigcup N_s \bigcup N_a$

K set of vehicles

Constants and parameters

τ time of the last routes determination or update

$t_{ij}(\tau)$ estimation of link travel time between nodes i and j at time τ

Q vehicle capacity

$q_i \leq Q$ demand of node i

$Q_k(\tau)$ load of vehicle k at time τ

e_i starting time of time window at node i

e_0 beginning of the planning period

l_i ending time of time window at node i

l_0 end of the planning period

$s_i \leq l_i - e_i$ service time at node i; $s_0 \equiv 0$

k_i vehicle that at time τ is located in node $i \in N_s \bigcup N_a$

M a very big positive number

Decision variables

x_{ijk}^t equals one if vehicle k departures from node i to node j at time t, otherwise equals zero

a_{ik} start of service at node i by vehicle k

5.2 Parameters initialisation

At every time moment τ when we construct vehicle routes for the first time or adjust them in response to some dynamic events, we have to initialise/update input parameters for the linear programming model. First of all, data sets have to be redefined. Set N_d of unserved demands contains only these demand nodes that have not yet been serviced by any vehicle. If at time τ a vehicle is waiting for service or serving a demand node i, then this node belongs to the set N_s. Please note that $N_d \cap N_s = \varnothing$. If a vehicle at time τ is in transit between two nodes, we create an artificial intermediate node $i \in N_a$ for it. In the next execution period, we consider the nodes from the set N_a as the starting nodes for the vehicles in transit. If a vehicle has just finished serving a client, we assume that it is already at the beginning of its journey to the next customer. Therefore, we create an artificial intermediate node for such vehicle as well. At the beginning of the planning period, when we create the routes for the first time and $\tau = e_0$, we set $N_s = N_a = \emptyset$.

Next, as the important input parameters for the model, we take the latest estimations of the link travel times $t_{ij}(\tau)$ as provided by the deployed navigation system. If they have changed considerably since the last routes adjustment, it may have a significant influence on the problem solution. Further, we redefine the parameters $Q_k(\tau)$, $k \in K$ and k_i, $i \in N_s \cup N_a$ which represent the load and number of vehicles at time τ and are known with certainty. Moreover, the time window boundaries for some nodes $i \in N_s$ have to be adjusted too. If $e_i < \tau$ then we move forwards the starting time of the corresponding time window and set $e_i = \tau$. In addition, we reduce the service time s_i by the amount of time that was used at the previous execution period. Besides this, for $i \in N_s$ we assume that the whole amount q_i should be unloaded in the next execution period.

5.3 Mathematical model

The time-dependent vehicle routing problem with time windows and online travel time information is formulated as a mixed-integer linear programming problem. The mathematical formulation is partly based on the network flow model of Toth and Vigo (2002, p.158):

$$\min \sum_{i \in N_{0dsa}} \sum_{j \in N_{0d}} \sum_{k \in K} \sum_{t \geq \tau} t_{ij}(\tau) x_{ijk}^t, \tag{1}$$

$$\sum_{j \in N_{0d}} \sum_{k \in K} \sum_{t \geq \tau} x_{ijk}^t = 1, \qquad i \in N_{dsa}, \tag{2}$$

$$\sum_{i \in N_{0dsa}} \sum_{t \geq \tau} x_{ijk}^t - \sum_{i \in N_{0d}} \sum_{t \geq \tau} x_{jik}^t = 0, \qquad j \in N_d, k \in K, \tag{3}$$

$$\sum_{j \in N_d} \sum_{t \geq \tau} x_{0jk}^t \leq 1, \qquad k \in K, \tag{4}$$

$$\sum_{i \in N_{0sa}} \sum_{j \in N_{0d}} \sum_{t \geq \tau} x_{ijk}^t - \sum_{i \in N_{dsa}} \sum_{t \geq \tau} x_{i0k}^t = 0, \qquad k \in K, \tag{5}$$

$$\sum_{j \in N_{0d}} x_{ijk_i}^{\tau} = 1, \qquad i \in N_a, \tag{6}$$

$$\sum_{j \in N_{0d}} x_{ijk_i}^{s_i} = 1, \qquad i \in N_s, \tag{7}$$

$$a_{ik} + s_i + t_{ij}(\tau) - a_{jk} \leq (1 - x_{ijk}^t) M, \qquad i \in N_{0dsa}, j \in N_{0d}, k \in K, t \geq \tau, \tag{8}$$

$$e_i \sum_{j \in N_{0d}} x_{ijk}^t \leq a_{ik}, \qquad i \in N_{ds}, k \in K, t \geq \tau, \tag{9}$$

$$a_{ik} \leq (l_i - s_i) \sum_{j \in N_{0d}} x_{ijk}^t, \qquad i \in N_{ds}, k \in K, t \geq \tau, \tag{10}$$

$$e_0 \leq a_{0k} \leq l_0, \qquad k \in K, \tag{11}$$

$$Q_k(\tau) + \sum_{i \in N_{ds}} q_i \sum_{j \in N_{0d}} \sum_{t \geq \tau} x_{ijk}^t \leq Q, \qquad k \in K, \tag{12}$$

$$x_{ijk}^t \in \{0,1\}, \qquad i \in N_{0dsa}, j \in N_{0d}, k \in K, t \geq \tau, \tag{13}$$

$$a_{ik} \geq 0, \qquad i \in N_{ds}, k \in K \tag{14}$$

The objective of this model is to minimise the total travel time over all routes. Restrictions (2)–(5) constitute the flow conservation constrains. Equation (2) states that each customer is visited by exactly one vehicle while equation (3) indicates that the same vehicle enters and leaves a given demand node. According to equations (4) and (5), each vehicle leaves depot at most once but if vehicle has left depot and is in transit, it must also return to it. Two next restrictions deal with vehicle starting times. In particularly, equation (6) specifies that vehicle k_i, that at time τ is located at the artificial intermediate node *i*, must immediately leave it. According to equation (7) vehicle k_i, that is serving or waiting for service at node *i,* must leave this node after the end of the service.

In order to guarantee schedule feasibility, the starting times of service at two connected nodes *i* and *j* must satisfy equation (8). Time windows restrictions are described by equations (9) and (10), where the first constrain prohibits start of service before the starting time of the corresponding time window and the second constrain forbids end of service after its closure. Restriction (11) indicates that vehicles may not leave the depot before the depot opening time and may not return to the depot after the depot closing time. Restriction (12) states that the cumulative load must not exceed the vehicle capacity. Finally, equations (13) and (14) are definitional constrains for binary flow variables and service starting time variables, respectively.

6 Solution method using a genetic algorithm

The time-dependent vehicle routing problem (1)–(14) is a generalisation of the classical Travelling Salesman Problem (TSP) and thus belongs to the class of NP-complete problems (Lenstra and Rinnooy Kan, 1981). Exact solution algorithms can solve to optimality only small instances of the problem, working unreasonably long for larger problems. Furthermore, the described time-dependent VRP is an asymmetric problem while a congestion may occur or a road may be blocked in only one direction, whilst the opposite direction of the same road remains free. In addition, the triangular property is not satisfied, while the direct link between two nodes may be hampered and a by-pass is a faster alternative even if one has to travel a longer distance. Hence, exact solution methods and classical heuristics cannot handle the formulated problem. Therefore, to solve it we implemented a genetic algorithm metaheuristics. Genetic algorithms were successfully deployed to VRPs and have proved to produce good quality solutions (e.g. Thangiah, 1995; Potvin and Bengio, 1996; Taniguchi and Shimamoto, 2004; Ombuki et al., 2006; Alvarenga et al., 2007; Hanshar and Ombuki-Berman, 2007; Marinakis et al., 2007, etc.). In particular, Hanshar and Ombuki-Berman (2007) showed that GA can deliver good results for a dynamic VRP, too.

6.1 Chromosome representation

To apply a genetic algorithm to any specific problem, first a chromosome representation should be chosen. In our approach, the information about the whole population is contained in one three-dimensional matrix. The first dimension of the matrix indexes various individuals of the population. The rows of the matrix (second dimension) correspond to different routes, whilst the columns (third dimension) correspond to the clients visited on each route. The number of routes for each individual, the number of clients per route, and the fitness values are stored in separate arrays. Such matrix representation is very convenient, as it allows quick and efficient sorting of the solutions within the population. This is possible, since the definition of the matrix through pointers does not require the overwriting of the elements; only redirection of the pointers should be made instead of this.

6.2 Initial population

To generate an initial population of feasible solutions we sort the customers by the starting time of the time window. The customer with the earliest starting time is taken as the first customer in the first route. Further customers are chosen randomly one after the other and appended to the route until the time schedule and capacity constraints are satisfied. If after hundred attempts no valid customer for the given route can be chosen, we initiate a new route. From the rest of the customers we again select the one with the earliest time window starting time and set this client as the first for the next route. The procedure is repeated until no unserved customers are left and enough individuals for the initial population are created.

6.3 Selection criteria

To select a set of parents for further reproduction, we implement a stochastic tournament selection operator (Dréo et al., 2006, p.88). The core of the operator is a tournament set which consists of k individuals randomly chosen from the population. In the deterministic tournament selection the fittest element from the tournament set is selected for the reproduction. In the stochastic variation of the operator, a random number $r \in [0, 1]$ is generated. If r is less than some number p, called selection pressure, then the fittest individual from the tournament set is selected for the reproduction. Otherwise, if $r \geq p$ then a random individual from the rest of the tournament set is selected. The selection pressure must be between 0.5 and 1 and shows the bias of the selection operator towards the fitter solution. It allows some less fit solutions to survive from generation to generation (Hanshar and Ombuki-Berman, 2007). The individuals that were chosen for the tournament set are replaced in the population, as this increases the variance of the process and favours the genetic drift. The pseudocode of the stochastic tournament selection algorithm is presented in Figure 2.

6.4 Crossover operator

We adopt the special crossover operator developed by Ombuki et al. (2006) that is particularly suitable for VRP with hard time windows. The authors carried experiments with two standard crossover operators, namely uniform order crossover and partially mapped crossover, and found the produced results unsatisfactory. Therefore, they introduced a new operator, called Best Cost Route Crossover (BCRC). The operator produces two feasible offspring from two parents p_1 and p_2 by executing the following procedure. In the first step, a random route is selected from each parent (route r_1 is selected from parent p_1 and r_2 from p_2). Then the customers that belong to the route r_2 are removed from the parent p_1. Analogously, the customers belonging to the route r_1 are removed from parent p_2. To yield the feasible children, the removed customers should be selected randomly and re-inserted back into the corresponding solution at the least cost. For that purpose the algorithm scans all possible locations for insertion and chooses the feasible ones. The removed customer is then inserted into the place that induces the minimum additional costs. If no feasible insertion place can be found, a new route containing the removed customer alone is created and added to the offspring. The pseudocode of the corresponding BCRC algorithm is listed in Figure 3. The proportion of the individuals that undergo the crossover operator is determined by a crossover rate; the rest survives intact.

Figure 2 Tournament selection algorithm

```
for (i = 1 to Population_Size)
    select randomly k individuals from the population //tournament set
    generate a random number r between 0 and 1
    if r < Selection_Pressure
        then select for reproduction the fittest individual
               from the tournament set
        else select for reproduction a random individual
              from the tournament set without the fittest element
end for
```

Figure 3 Best cost route crossover algorithm

```
for (i = 1 to Crossove_Rate × Population_Size)
    select randomly two parents p1 and p2
         // remove the parents from the reproduction set
    select randomly one route from each parent, r1 and r2
    remove from p1 customers that belong to r2
    remove from p2 customers that belong to r1
    for every customer in r2 reinsert it into p1 at the best (least cost) place
    for every customer in r1 reinsert it into p2 at the best (least cost) place
         // if for some customer its reinsertion is infeasible,
         // create a separate route for it
end for
```

6.5 Mutation operator

Finally, a mutation operator is applied to the population to ensure that the algorithm does not converge prematurely to a local optimum. As mutation introduces a random alteration to diversify the search, it can be a relatively destructive element, deteriorating the fitness of the solution. Therefore, the mutation operator is applied to only small fraction of the offspring, determined by the mutation rate. We applied a widely-used swap mutation algorithm, exchanging two customers with similar time windows. The two windows are considered to be similar if the difference between their starting times is the smallest (Alvarenga et al., 2007). The swap operator proceeds as follows: A random customer is selected from a random tour. Then a customer with the similar time window is searched for and the two customers are exchanged. If the obtained solution is infeasible, the procedure is repeated. The pseudocode of the swap mutation operator is provided in Figure 4.

Figure 4 Mutation operator

```
for (i = 1 to Population_Size)
    generate a random number r between 0 and 1
    if r < Mutation_Rate then
        repeat
            select a random customer from a random route
            search for a customer from other routes with the similar time
window
            swap the two customers
        until a feasible solution created
    end if
end for
```

6.6 Construction of a new population

In the new generation, the offspring created by the sequential application of the selection, crossover and mutation operators completely replace the parents. Only the small numbers of the worst offspring are left aside and instead of them the best individuals from the old generation, called *elite,* are included into the new generation. It ensures that the best solutions can propagate through generations without the effect of the crossover or mutation operators. Therefore, the fitness value of the best solution is monotonically non-decreasing from one generation to another (Dréo et al., 2006, p.91). In the new generation, however, the *elite* individuals have to compete with the fittest offspring, forcing the algorithm to converge towards an optimum.

7 Computational results

7.1 Constant travel time tests

The proposed genetic algorithm was tested in two stages: Stage one with constant travel times and stage two with variable travel times. Even though the considered problem is dynamic and time-dependent, the algorithm was initially tested on constant travel time data to prove its efficiency. For this purpose, we take the Solomon's benchmark problems with the long scheduling horizon, namely the sets R2, C2 and RC2 (Solomon, 1987). Test problems with the narrow horizon were discarded as during one trip a vehicle can serve only few customers. So, they are improper for the dynamic problem at hand. The algorithm was coded in C and run on a 3 GHz Intel Pentium IV machine with 1GB memory running SuSE Linux 10.2.

The results for the R2 and RC2 problems are presented in Table 1. Column 'Average TT' indicates average travel time calculated over ten runs for each problem instance, whilst column 'Best TT' contains the best found solution. The two columns 'Best known' state the total distance and the number of used vehicles for the best known published solutions identified by heuristic methods.[1] Even though these numbers indicate the shortest travelled distance and we are minimising the total travel time, this fact does not influence the results, as at the moment we consider constant travel time problems and set the vehicle speed equal to one. Hence the travelled time equals the distance. As benchmark solutions we take the ones produced by heuristics, not by exact algorithms, while till these days not all Solomon problems are solved to optimality. Furthermore, it is not always correct to compare exact methods, which produce optimal solutions but need considerable amount of computational time, and heuristics, which are designed to produce fast solutions of a good quality.

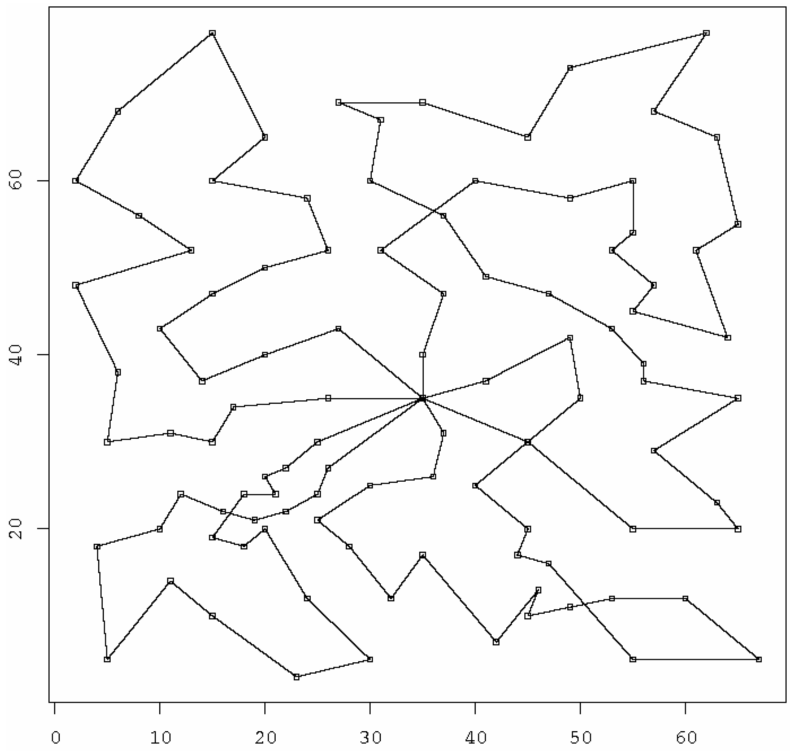
Table 1 Test results for constant travel times

Problem	*Genetic algorithm*			*Best known*		*Quality in%*
	Average TT	*Best TT*	*NV*	*TD*	*NV*	
R201	1200.82	1189.28	8	1165.3	8	2.06
R202	1083.56	1073.69	7	1053.9	6	1.88
R203	910.71	**897.47**	6	910.1	5	–1.39
R204	751.39	**739.23**	4	770.8	4	–4.10
R205	984.95	975.01	5	972.5	5	0.26
R206	907.44	**897.83**	5	906.1	3	–0.91
R207	820.64	**804.35**	4	860.1	4	–6.48
R208	720.98	**709.40**	4	744.1	4	–4.66
R209	883.59	**867.26**	5	888.5	5	–2.39
R210	933.94	**921.43**	6	939.3	3	–1.90
R211	777.45	**763.37**	4	815.5	5	–6.39
RC201	1309.75	1278.69	8	1274.3	8	0.34
RC202	1130.18	**1113.05**	7	1119.5	7	–0.58
RC203	976.22	967.82	5	958.0	5	1.02
RC204	807.81	**796.58**	4	798.4	3	–0.23
RC205	1170.81	1161.55	7	1154.0	7	0.65
RC206	1083.47	**1062.75**	6	1080.4	6	–1.63
RC207	1000.79	**974.50**	5	1005.6	7	–3.09
RC208	805.71	**790.86**	5	820.5	5	–3.61

Note: TT: travel time, NV: number of vehicles, TD: total distance.

The efficiency of the developed genetic algorithm is presented in column ‘Quality’. The quality of our solution is defined as the relative deviation from the corresponding best known solution. The negative value means that our solution is better than the corresponding best known, while the positive value indicates that our solution is worse. As can be seen from the table, the received results are comparable with best known so far. In fact, for eight instances of R2 problems and five instances of RC2 problems we were able to outperform the best known solutions (emphasised in bold). Figure 5 illustrates the network topology for the received solutions by the example of R211 problem with random geographic data. Concerning the C2 problem set, the produced results are quite unsatisfactory while the developed algorithm converges prematurely to a local optimum. Presumably, it is not suitable for clustered problems especially with tight time windows, or some other sets of genetic algorithm parameters have to be chosen.

Figure 5 Best found solution for R211 problem



7.2 Test results for real-time travel times

The results from the previous section show that the developed genetic algorithm has proved to be efficient for constant travel times and thus can be further used in the second stage of our simulation study with real-time travel times. In this stage, we assume that travel times between a pair of nodes undergo two types of disturbances. On the one hand, a link travel time function depends on the time of day when a vehicle drives along this link. Thus we capture time dependency due to periodic traffic congestions that is based on historic data and hence known a priori. On the other hand, we incorporate unpredicted short-term fluctuations of travel times that occur due to unexpected dynamic events like accidents.

In practice, actual travel times are provided by an advanced traveller information system. For our experiment, however, we have to simulate them. Similar to other authors (e.g. Chen et al., 2006; Donati et al., 2008) we divide the planning period into four intervals and define travel times as a step function. To introduce the time-dependency of travel times, we multiply the original values by the coefficients 1.1, 0.9, 1.2 and 0.8, respectively. The changeovers between intervals are smoothed in order to satisfy the FIFO condition. The short-term fluctuations of travel times caused by accidents or other traffic impediments are modelled through random perturbations. They are set equal to the absolute value of a normal random variable with mean $\mu = 0$ and variation $\sigma = 9$ (cf. Potvin et al., 2006).

Further we have to chose how often to update travel times. According to Montemanni et al. (2005), who carried out extensive tests on both artificial and real data, the best value for the time slice equals approximately 20 minutes. (The authors consider an eight-hour working day and divide it into 25 time intervals.) One should not take a very short time slice, while the routes re-optimisation and driver notification would be undertaken too often. On the other hand, very large time intervals are also inappropriate. The vehicles would follow less-than-optimal routes determined for already obsolete travel times, while the most recent data about traffic conditions would be ignored. We believe that 10 or 15 minutes is the absolute minimum for the duration of a time slice. The maximal duration depends on the range and degree of travel time fluctuations and can vary considerably for different settings.

To simulate the vehicle routing in the dynamic and time-dependent case, we deploy the concept of events manager (Montemanni et al., 2005; Hanshar and Ombuki-Berman, 2007). The events manager is a separate module that serves as an interface between the external world and the optimisation procedure and is responsible for routes adjustments caused by changes in travel times. When a new dynamic event arrives, the events manager creates a static problem and runs the genetic algorithm to solve it. As inputs the events manager takes the list of not yet serviced customers, the latest travel time estimations and the current locations of vehicles. As output, it returns the vehicle routes that are the best found subject to dynamic travel times. Afterwards, these routes are transmitted to drivers and anticipated arrival times are communicated to customers. The pseudocode of the events manager module is presented in Figure 6.

The test results for the real-time case are presented in Table 2. Column 'Updated routes' contains the average travel time value calculated over 20 runs when the routes re-optimisation was undertaken after every perturbation of the travel time matrix. On the contrary, column 'Initial routes' states the results for the case when travel times are periodically updated but the routes are not correspondingly adjusted. Consequently, the vehicles have to follow the initial routes constructed at the beginning of the planning period. The value difference between the two columns shows that even for small perturbations of the travel times the periodic routes adjustment leads to better results.

Figure 6 Events manager

Time = 0
create a list of unserved customer orders
get the latest estimations for the travel times
set vehicle initial locations to depot
repeat
  create a static problem
  run the genetic algorithm
  for (*each route from the best found solution*)
    remove from the unserved list customers that have been served till the moment Time + Δt
    update vehicle capacities
    calculate the vehicle locations at the moment Time + Δt
    create artificial intermediate nodes for vehicles being on road at Time + Δt
    sum up the travelled till Time + Δt *time*
  end for
  update travel times:
    → *integrate artificial intermediate nodes*
    → *multiply travel times by the corresponding coefficient*
    → *apply random perturbations*
  update list of rejected customers
  Time = Time + Δt
until *no unserved customers left*

Table 2 Test results for real-time travel times

Problem	*Updated routes*	*Initial routes*	*Rejected in%*	*Problem*	*Updated routes*	*Initial routes*	*Rejected in%*
R201	1211.06	1218.51	15	RC201	1319.02	1320.84	0
R202	1084.57	1111.92	20	RC202	1148.95	1168.27	5
R203	910.07	929.85	0	RC203	987.44	995.53	20
R204	759.15	757.26	5	RC204	817.23	829.63	10
R205	1003.64	1022.87	10	RC205	1197.12	1179.83	0
R206	910.32	915.43	15	RC206	1098.97	1130.09	15
R207	831.21	836.95	15	RC207	1021.00	1021.01	10
R208	731.18	732.54	20	RC208	825.43	834.00	5
R209	890.04	896.07	10				
R210	948.58	956.41	5				
R211	794.88	811.11	10				

Finally, column 'Rejected' indicates the fraction of problem instances containing customers that could not be served in the case without route re-optimisation. (In the case when route re-optimisation was undertaken after every update of the travel time matrix, all customers could be served.) The customers must be rejected while the traversed routes are definitely less-than-optimal while being determined for obsolete travel times. Hence, the vehicles arrive to the customers after the ending time of the time windows and are not able to serve them. This result supports the findings of Fleischmann et al. (2004) who claim that the usage of constant travel times underestimates the route duration of about 10%. Consequently, when solutions for the constant travel time case are deployed in real-time settings, their optimality and even feasibility are subject to substantial changes. Therefore, to be able to serve all customers and decrease costs one has to make use of modern technologies and real-time data and promptly react to ever-changing settings of the real world.

8 Conclusions

The paper deals with a vehicle routing problem with real-time travel times. In particularly, we consider a case when there is a mobile connection between the drivers and the dispatching centre that allows communicating them new instructions to action in real time. Furthermore, we assume that the central office makes use of the advanced technologies to gain an overview over variable traffic conditions and locations of vehicles on road. This fact allows us to investigate time-dependent travel times which are updated on a permanent basis. Thus we consider the possibility to react to unpredicted traffic impediments and divert a vehicle en route from its current destination. For the given problem we formulated a mixed-integer linear programming model and developed a genetic algorithm to solve it. We performed an extensive computational study in order to prove the efficiency of the proposed algorithm on the well-known static benchmarks and to test its performance in dynamic settings. The achieved results are competitive with the best published solutions.

The computational study that incorporates dynamic or stochastic customers is postponed for future research. Furthermore, we plan to improve our model by developing a mechanism that stores elements of best solutions computed for one time slice and takes this information as an input in the subsequent time slices. Currently for every time slice the problem is solved from scratch. However, if between time intervals only small changes of input parameters take place, then the corresponding solutions should not differ much. Therefore, some sort of long-term memory that contains previous good solutions or their elements should save computational time and increase efficiency of the algorithm. What is more, it would be interesting to integrate into the model statistical data about past customer requests in order to forecast the distribution of future orders in time and space. And finally, testing the developed framework in real-life settings with authentic customer orders and travel times would be also of great interest.

References

Alvarenga, G.B., Mateus, G.R. and de Tomi, G. (2007) 'A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows', *Computers & Operations Research*, Vol. 34, pp.1561–1584.

Attanasio, A., Cordeau, J-F., Ghiani, G. and Laporte, G. (2004) 'Parallel tabu search heuristic for the dynamic multi-vehicle dial-a-ride problem', *Parallel Computing*, Vol. 30, pp.377–387.

Bertsimas, D.J. and van Ryzin, G. (1991) 'A stochastic and dynamic vehicle routing problem in the Euclidean plane', *Operations Research*, Vol. 39, pp.601–615.

Bertsimas, D.J. and van Ryzin, G. (1993) 'Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles', *Operations Research*, Vol. 41, pp.60–76.

Cetin, M. and List, G.F. (2006) 'Integrated modeling of information and physical flows in transportation systems', *Transportation Research Part C*, Vol. 14, pp.139–156.

Chen, H-K., Hsueh, C-F. and Chang, M-S. (2006) 'The real-time time-dependent vehicle routing problem', *Transportation Research Part E*, Vol. 42, pp.383–408.

Dantzig, G.B. and Ramser, J.H. (1959) 'The truck dispatching problem', *Management Science*, Vol. 6, pp.80–91.

Donati, A.V., Montemanni, R., Casagrande, N., Rizzoli, A.E. and Gambardella, L.M. (2008) 'Time dependent vehicle routing problem with a multi ant colony system', *European Journal of Operational Research*, Vol. 185, pp.1174–1191.

Dréo, J., Pétrovski, A., Siarry, P. and Taillard, E. (2006) *Metaheuristics for Hard Optimization*, Springer, Berlin.

Fleischmann, B., Gietz, M. and Gnutzmann, S. (2004) 'Time-varying travel times in vehicle routing', *Transportation Science*, Vol. 38, pp.160–173.

Gendreau, M. and Potvin, J-Y. (1998) 'Dynamic vehicle routing and dispatching', in Crainic, T.G. and Laporte, G. (Eds): *Fleet Management and Logistics*, Kluwer, pp.115–126.

Ghiani, G., Guerriero, F., Laporte, G. and Musmanno, R. (2003) 'Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies', *European Journal of Operational Research*, Vol. 151, pp.1–11.

Golden, B.L. and Assad, A.A. (1988) *Vehicle Routing: Methods and Studies*, North-Holland.

Golden, B.L., Wasil, E.A., Kelly, J.P. and Chao, I-M. (1998) 'Metaheuristics in vehicle routing', in Crainic, T.G. and Laporte, G. (Eds): *Fleet Management and Logistics*, Kluwer, pp.33–56.

Haghani, A. and Jung, S. (2005) 'A dynamic vehicle routing problem with time-dependent travel times', *Computers & Operations Research*, Vol. 32, pp.2959–2986.

Hanshar, F.T. and Ombuki-Berman, B.M. (2007) 'Dynamic vehicle routing using genetic algorithms', *Applied Intelligence*, Vol. 27, pp.89–99.

Ichoua, S., Gendreau, M. and Potvin, J-Y. (2003) 'Vehicle dispatching with time-dependent travel times', *European Journal of Operational Research*, Vol. 144, pp.379–396.

Jung, S. and Haghani, A. (2001) 'Genetic algorithm for the time-dependent vehicle routing problem', *Transportation Research Record 1771*, pp.164–171.

Küpper, A. (2005) *Location-Based Services: Fundamentals and Operation*, Wiley, Chichester, England.

Laporte, G. (1992) 'The vehicle routing problem: an overview of exact and approximate algorithms', *European Journal of Operational Research*, Vol. 59, pp.345–358.

Laporte, G., Gendreau, M., Potvin, J-Y. and Semet, F. (2000) 'Classical and modern heuristics for the vehicle routing problem', *International Transactions in Operational Research*, Vol. 7, pp.285–300.

Lenstra, J.K. and Rinnooy Kan, A.H.G. (1981) 'Complexity of vehicle routing and scheduling problems', *Networks*, Vol. 11, pp.221–227.

Malandraki, C. (1989) *Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computational Experiments*, PhD Thesis, Northwestern University, Evanson.

Malandraki, C. and Daskin, M.S. (1992) 'Time dependent vehicle routing problems: formulations, properties and heuristic algorithms', *Transportation Science*, Vol. 26, pp.185–200.

Malandraki, C. and Dial, R.B. (1996) 'A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem', *European Journal of Operational Research*, Vol. 90, pp.45–55.

Marinakis, Y., Migdalas, A. and Pardalos, P.M. (2007) 'A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm', *Journal of Global Optimization*, Vol. 38, pp.555–580.

Mitrović-Minić, S., Krishnamurti, R. and Laporte, G. (2004) 'Double-horizon based heuristic for the dynamic pickup and delivery problem with time windows', *Transportation Research Part B*, Vol. 38, pp.669–685.

Montemanni, R., Gambardella, L.M., Rizzoli, A.E. and Donati, A.V. (2005) 'Ant colony system for a dynamic vehicle routing problem', *Journal of Combinatorial Optimization*, Vol. 10, pp.327–343.

Ombuki, B., Ross, B.J. and Hanshar, F. (2006) 'Multi-objective genetic algorithms for vehicle routing problem with time windows', *Applied Intelligence*, Vol. 24, pp.17–30.

Potvin, J-Y. and Bengio, S. (1996) 'The vehicle routing problem with time windows – Part II: genetic search', *INFORMS Journal of Computing*,Vol. 8, pp.165–172.

Potvin, J-Y., Xu, Y. and Benyahia, I. (2006) 'Vehicle routing and scheduling with dynamic travel times', *Computers & Operations Research*, Vol. 33, pp.1129–1137.

Psaraftis, H.N. (1988) 'Dynamic vehicle routing problems', in Golden, B.L. and Assad, A.A. (Eds): *Vehicle Routing: Methods and Studies*, North-Holland, pp.223–248.

Psaraftis, H.N. (1995) 'Dynamic vehicle routing: status and prospects', *Annals of Operations Research*, Vol. 61, pp.143–164.

Solomon, M.M. (1987) 'Algorithms for the vehicle routing and scheduling problems with time window constrains', *Operations Research*, Vol. 35, pp.254–265.

Swihart, M. and Papastavrou, J.D. (1999) 'A stochastic and dynamic model for the single-vehicle pick-up and delivery problem', *European Journal of Operational Research*, Vol. 114, pp.447–464.

Taniguchi, E. and Shimamoto, H. (2004) 'Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times', *Transportation Research Part C*, Vol. 12, pp.235–250.

Thangiah, S.R. (1995) 'Vehicle routing with time windows using genetic algorithms', in Chambers, L. (Ed.): *Applications Handbook of Genetic Algorithms: New Frontiers*, Vol. 2, CRC Press, Boca Raton, FL, pp.253–277.

Toth, P. and Vigo, D. (Eds) (2002) *The Vehicle Routing Problem, Society for Industrial and Applied Mathematics*, Philadelphia.

Notes

1 The prevailing majority of best-known solutions identified by heuristics are taken from the article by Alvarenga et al. (2007). The results for the problems R206, R210 and RC204 are taken from the site http://w.cba.neu.edu/~msolomon/problems.htm (accessed on 21 February 07).