The Real-Time Vehicle Routing Problem

Irena Okhrin and Knut Richter

Department of Industrial Management, European University Viadrina
Grosse Scharrnstrasse 59, 15230 Frankfurt (Oder), Germany
irena.okhrin@euv-frankfurt-o.de, richter@euv-frankfurt-o.de

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 minimize operational costs. In the past, vehicles executing routes and dispatchers in the control center 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. 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.

2 Problem Description

We consider a vehicle routing problem with online travel time information. The problem is defined on a complete graph $G = (V, A)$, where V is the vertex set and A the arc set. Vertex 0 represents a depot whilst other vertexes 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 minimized.

The frequently considered constant travel time function is not realistic. In practice, it 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 it may be diverted from its route. 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, instead of deviating from its route and serving other customers in the meantime.

Thanks to mobile technology we can overcome the mentioned shortcomings and model vehicle routing in more realistic settings. State-of-the-art mobile technologies substantially facilitate the dynamic vehicle routing. First, they allow locating vehicles in real time. This gives the decision center the overview over the routes execution. Second, they enable the online communication between the drivers and the dispatching center. Thus, new instructions can be sent to drivers at any time, regardless of their location and status. And finally, mobile technologies are capable to capture varying traffic conditions in real time and in the short run predict with high accuracy the travel time between a pair of nodes. All these factors enable modelling approaches that even better approximate the real-world conditions.

We formulate the real-time VRP as a series of static vehicle routing problems with heterogeneous fleet at a specific point of time. We use the concept of time rolling horizon and run a re-optimization procedure to find new vehicle routes every time when the travel time between a pair of nodes is updated. For the vehicles that at time of routes adjustment are in transit to their destinations we create artificial intermediate nodes. The re-optimization algorithm is then performed on the graph that includes also the artificial intermediate nodes.

3 Solution Algorithm

The time-dependent vehicle routing problem is a generalization of the classical travelling salesman problem (TSP) and thus belongs to the class of NP-complete problems [4]. Exact solution algorithms can solve to optimality only small instances of the problem, working unreasonably long for larger problems. Hence, to solve the described problem we implemented a genetic algorithm metaheuristics. Genetic algorithms were successfully deployed to VRPs and have proved to produce good quality solutions [1, 3, 5, 8].

Initial population. Unlike many heuristics, a genetic algorithm works with groups of solutions, instead of considering one solution at the time. Therefore, an initial population of feasible solutions has to be generated at the beginning of the algorithm performance. We develop a fast and effective method to initially assign all customers to routes. At first 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 constrains 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.

Selection criteria. To select a set of parents for further reproduction we implement the stochastic tournament selection operator [2, p. 88]. The core of the operator is a tournament set which consists of k individuals randomly chosen from the population. These individuals are replaced in the population, what increases the variance of the process and favours the genetic drift.

Crossover operator. We adopt the special crossover operator called best cost route crossover (BCRC) [8], which is particularly suitable for VRP with hard time windows. 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.

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 [1].

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 number 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. Such strategy is called elitism [6, p. 168]. 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 nondecreasing from one generation to another [2, 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.

4 Computational Results

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 the constant travel time data to prove its efficiency. For this purpose we take the Solomon's benchmark problems with the long scheduling horizon [9]. The received results for the constant travel time tests are comparable with best known so far. In fact, for eight instances out of eleven from the random problem set and for five instances out of eight from the semi-clustered problem set we were able to outperform the best known solutions. For more details about constant travel time test please see [7].

The second stage of the computational experiments simulates the vehicle routing in more realistic settings. Here 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 which is based on historic data and hence known a priori. On the other hand, we incorporate unpredicted short-

Table 1. Test results for real-time travel times

Problem	Average with re-optim.	Average without re-optim.	Rejected in %	Problem	Average with re-optim.	Average without re-optim.	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				

term fluctuations of travel times that occur due to unexpected dynamic events like accidents. Therefore, we assume that the dispatching centre has the real-time overview over traffic conditions. Based on these data it updates the optimal solution and periodically adjusts the vehicle routes in order to avoid hampered roads.

The test results for the real-time case are presented in Table 1. Column “Average with re-optim.” contains average travel time value calculated over twenty runs when routes re-optimization was undertaken after every perturbation of the travel time matrix. On the contrary, column “Average without re-optim.” 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 travel times the periodic route adjustment leads to better results. Finally, column “Rejected” indicates the fraction of problem instances containing customers that could not be served in the case without route re-optimization. This is due to the fact that 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. From these experiments we can see that if solutions computed for constant travel times 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 the ever-changing settings of the real world.

5 Conclusions

The paper deals with a vehicle routing problem with real-time travel times. We assume the deployment of mobile information and communication system that allows us to consider time-dependent travel times which are updated on a permanent basis. Thus we incorporate the possibility to react to traffic impediments and divert a vehicle en route from its current destination. To solve the developed problem we implement a genetic algorithm. We perform an extensive computational study in order to prove the efficiency of the proposed algorithm on well-known static benchmarks as well as to test its performance in dynamic settings. The achieved results are competitive with best published solutions and prove the efficiency of the proposed solution method.

References

1. Alvarenga GB, Mateus GR, de Tomi G (2007) A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. Computers & Operations Research 34:1561–1584
2. Dréo J, Pétrovski A, Siarry P, Taillard E (2006) Metaheuristics for Hard Optimization. Springer
3. Hanshar FT, Ombuki-Berman BM (2007) Dynamic vehicle routing using genetic algorithms. Applied Intelligence 27:89–99
4. Lenstra JK, Rinnooy Kan AHG (1981) Complexity of vehicle routing and scheduling problems. Networks 11:221–227
5. Marinakis Y, Migdalas A, Pardalos PM (2007) A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. Journal of Global Optimization 38:555–580
6. Mitchell M (1996) An Introduction to Genetic Algorithms. MIT Press
7. Okhrin I, Richter K (2007) The vehicle routing problem with real-time travel times. Arbeitsberichte Mobile Internet Business 8, European University Viadrina, Frankfurt (Oder)
8. Ombuki B, Ross BJ, Hanshar F (2006) Multi-objective genetic algorithms for vehicle routing problem with time windows. Applied Intelligence 24:17–30
9. Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constrains. Operations Research 35:254–265