Journal of Technical Science & Technologies; ISSN: 2298-0032; e-ISSN: 2346-8270; Volume 5, Issue 2, 2016

Real-World Applications of the Vehicle Routing Problem in Georgia

Irakli RODONAIA*
Artioma MERABIANI**

Abstract

This study will examine the value of real-time traffic information to optimal vehicle routing in a non- stationary stochastic network. The goal is to find a systematic approach to aid in the implementation of transportation systems integrated with real-time information technology. Finding the way to develop decision making procedures for determining the optimal driver attendance time, optimal departure times, and optimal routing policies is the primary goal. With studies based on a road network in Georgia, we aim to reduce vehicle usage while satisfying or improving service levels for just-in-time delivery.

Keywords: vehicle routing problem, traffic congestion reduction, optimization

Introduction

Real-world applications of the vehicle routing problem, in contrast with its canonical definition, often include two important dimensions: *evolution* and *quality of information.* Evolution of information is related to the fact that in some problems the information available to the planner may change during the execution of the routing, for instance, with the arrival of new client requests or changes in the travelling times. Quality of information reflects possible uncertainty on the available data, for example, when the demand of a client is only known as an estimation of its real demand (Pillac, 2011). A number of technological advances have increased the importance of *online or real-time applications.* With the introduction of the *Global Positioning System (GPS)* in 1996, the development and widespread use of cell-phones and smart-phones, combined with accurate *Geographic Information Systems (GIS),* companies are now able to track and manage their fleet in real-time cost effectively. Unlike traditional two-step process, vehicle routing can now be done dynamically, introducing greater opportunities to reduce operation costs, improve customer service, and reduce environmental impact.

The most common source of dynamism in vehicle routing is the online arrival of customer requests during the operation. More specifically, a request is generally composed of a location and a demand (Pillac, 2011). In our approach a travel time, a dynamic component of the most real-world applications is especially taken into account, while service time, that is, the time spent at the location of a client, is not explicitly studied but remains considered in most approaches. Finally, some works consider vehicle availability with the source of dynamism being the possible breakdown of vehicles.

By their nature, dynamic routing problems differ from their static equivalent by adding more degrees of freedom for the decision making and introducing new metrics for the objective function. In some contexts, such as the pick-up of express courier, the transport company does not have obligation to service a customer request. As a consequence, it can reject a request, either because it is simply impossible to serve it, or because the cost of serving is too high compared with the company objectives. Dynamic problems also frequently differ from their static counterparts in their objective function. In particular, while a common objective in static context is to minimize a routing cost, dynamic routing may introduce other notions such as service level, throughout or revenue maximization. Having to answer dynamic customer requests we introduce the notion of response time: a customer might request to be served as soon as possible, in which case the main objective may become to minimize the time between a request and its service (Pillac, 2011).

In dynamic problems critical information is revealed over time, meaning that the complete definition of an instance of a given problem is only known at the end of the planning horizon. As a consequence, an optimal solution can only be found a-posteriori. Therefore, most approaches use fast

*Prof.Dr., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia.
E-mail: irakli.rodonaia@ibsu.edu.ge
**MSc., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia.
E-mail: merabianiartioma@ibsu.edu.ge

approximation methods (so called *meta-heuristics*) that give a good solution in a relatively low computational time, rather than exact methods (dynamic programing, linear programing, Markov processes, etc.) that would only provide an optimal solution for the current state, providing no guarantee that the solution will be optimal once new data becomes available.

As it was stated above, travel time is considered as the problem of great importance and it should be taken into account to get more practical and reliable solutions. Due to a growing amount of traffic and a limited capacity of the road network, traffic congestion has become a daily phenomenon. Since *traffic congestion* causes heavy delays, it is very costly for intensive road users such as logistic service providers and distribution firms. In particular, such delays cause large costs for hiring the truck drivers and the use of extra vehicles, and if they are not accounted for in the vehicle route plans they may cause late arrivals or even violations of driving hour's regulations. Therefore, accounting for traffic congestion has a large potential for cost savings (Kok, et al., 2013). Travel time delays are mostly attributable to the so called 'recurrent' congestion that, for example, develops due to high volume of traffic seen during peak commuting hours. Incidents, such as accidents, vehicle breakdowns, bad weather, work zones, lane closures, special events, etc. are other important sources of traffic congestion. This type of congestion is labeled '*non-recurrent*' congestion in that its location and severity is unpredictable. It is reported that over 50% of all travel time delays are attributable to the non-recurrent congestion (Guner et al. 2012). As it was said, presence of congestions and their time characteristics are not known beforehand, but are revealed only when the realization of an initial routing plan is started. When a vehicle chooses a link (road segment) to traverse, there is a fixed probability that it will actually traverse an adjacent link as opposed to the one chosen. A path from the source to the destination is chosen a priori. Then there is a fixed probability upon arriving to one of the nodes in the network that the next link is congested and an alternate route should be chosen.

In the paper, we develop a combined approach that takes into account all abovementioned issues in the conditions of road networks of Georgia. To reflect real-world requirements the approach has two stages. In the first stage an initial (comprehensive) solution (routing) for all depot and all vehicles is obtained. In the second stage, a set of partial solutions for each vehicle is developed. The second stage is adaptive and essentially deals with dynamic real-time traffic information (including congestions). The result of the second stage is rerouting for each vehicle based on the real-time conditions and some optimality principles.

Formal problem definition

To solve the first stage problem, let us consider a unified heuristic which is able to solve different variants of the vehicle routing problem: the vehicle routing problem with time windows (VRPTW), the multi-depot vehicle routing problem (MDVRP), etc. (Pisinger, et al., 2005). All problem variants are transformed to a rich pickup and delivery model and solved using the Adaptive Large Neighborhood Search (ALNS) framework. The ALNS framework is an extension of the Large Neighborhood Search framework with an adaptive layer. All problem types are transformed to a Rich Pickup and Delivery problem with time windows (RPDPTW) and are solved using the Adaptive Large Neighborhood Search (ALNS) framework presented in (Stefan Ropke, 2009). In the RPDPTW we have a number of requests to be carried out by a fixed set of vehicles. Each request consists of picking up a quantity of goods at one location and delivering it to another location. The objective of the problem is to find a feasible set of routes for the vehicles so that all requests are serviced the overall travel distance is minimized. A feasible route of a vehicle should start at a given location, service a number of requests in a way that the capacity of the vehicle is not exceeded, and finally end at a given location. A pickup or delivery should take place within a given time window. Each request, furthermore, has an associated pickup precedence number and a delivery precedence number. A vehicle must visit the locations in non-decreasing order of precedence's. Since not all vehicles may be able to service all requests (e.g. due to their physical size or the absence of some cooling compartments) we need to ensure that every request is serviced by a given subset of vehicles. Between any two locations we have an associated, nonnegative distance and travel time.

In order to transform the vehicle routing problem with time windows (VRPTW) instance to a RPDPTW instance we map every customer in the VRPTW to a request in the RPDPTW. Such a request consists of a pick up at the depot and delivery at the customer site. The amount of goods that should be carried by the requests is equal to the demand of the corresponding customer. The time window of the pickup is set to $[a_d, a_d]$ where ad is the start of the time window of the depot in the VRPTW and its service time is set to zero. The time window and service time of the delivery are copied from the corresponding customer in the VRPTW.

As for the multi-depot vehicle routing problem (MDVRP), we cannot use these depots to model the MDVRP. The problem is that we should assign each pickup to a depot and we do not know which depot is going to serve a given request. Instead we create a dummy base location where all routes start and end and where all ordinary requests are picked up. We also create a dummy request for each vehicle k in the problem. The pickup and delivery locations of these requests are located at the depot of the corresponding vehicle. A dummy request has demand zero, it does not have any service time and it can be served at any time. The set Nk of each vehicle k contains all ordinary requests and the dummy request corresponding to the vehicle. This way, we ensure that each vehicle will carry precisely one dummy request.

The general idea of the ALNS framework (Pisinger, 2005) is to repeatedly remove requests from the solution and to reinsert them at a more profitable position. This is done by special destroy and repair heuristics. In each iteration, a destroy and a repair heuristic are chosen and applied. The selection is based on the past success of the heuristics. Compared to many local search heuristics that only apply very small changes to a solution, ALNS works

Journal of Technical Science & Technologies; ISSN: 2298-0032; e-ISSN: 2346-8270; Volume 5, Issue 2, 2016

with a larger search space, the so-called neighborhood N of the current solution. A neighborhood N(s) of the solution s is a set of solutions. The neighborhood contains all the solutions that could be created by changing that part of the solution. The term *neighbor* refers to the similarity between the solution s and its neighbors in N(s). A distance measure can be applied to the solutions in the search space, e.g. the Hamming distance. Solutions in N(s) usually have a comparatively low distance to s (Lutz, 2014). The example of destroy and repair is shown in Fig. 1. A destroy method in Fig. 1 could remove, say 15%, of the customers in the current solution, shortcutting the routes where customers have been removed. A very simple destroy method would select the customers to remove at random. A repair method could rebuild the solution by inserting removed customers, using a greedy heuristic. Such a heuristic could simply scan all free customers, insert the one whose insertion cost is the lowest and repeat inserting until all customers have been inserted.

ALNS can be based on any local search framework, e.g. simulated annealing, tabu search, guided local search (Pisinger et al., 2005). The general framework is outlined in Fig.2. Implementing a simulated annealing algorithm is straightforward as one solution is sampled in each iteration of the ALNS. A simple tabu search could, for example, be implemented by randomly sampling a number of candidate solutions and choosing the best non tabu solution. Several variables are maintained by the algorithm (Pisinger, et al., 2010). The variable xb is the best solution observed during the search, x is the current solution and x_t is a temporary solution that can be discarded or promoted to the status of current solution.

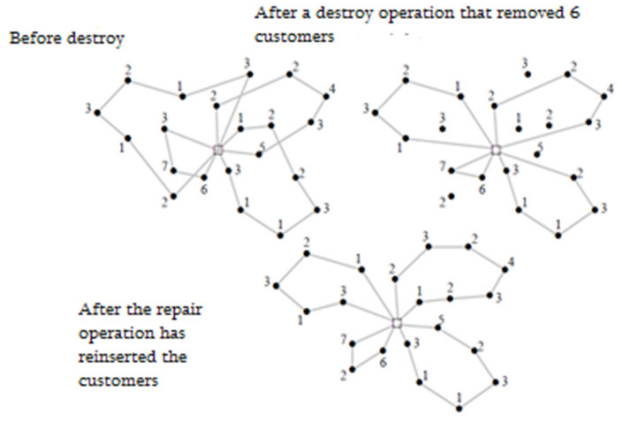

Figure 1. *Destroy and Repair principle*

The sets of destroy and repair methods are denoted W^- and W^+, respectively. Two variables are introduced in line 2:

$r-\subset R|W^-|$ and $r+\subset R|W^+|$, to store the weight of each destroy and repair method, respectively. Initially all methods have the same weight.

The function $d(\cdot)$ is the destroy method while $r(\cdot)$ is the repair method. More specifically, $d(x)$ returns a copy of x that is partly destroyed. Applying $r(\cdot)$ to a

Algorithm Adaptive large neighborhood search

1: input: a feasible solution x
2: $x^b = x; \rho^- = (1,\ldots,1); \rho^+ = (1,\ldots,1);$
3: **repeat**
4: select destroy and repair methods $d \in \Omega^-$ and $r \in \Omega^+$ using ρ^- and ρ
5: $x^t = r(d(x));$
6: **if** accept(x^t, x) **then**
7: $x = x^t;$
8: **end if**
9: **if** $c(x^t) < c(x^b)$ **then**
10: $x^b = x^t;$
11: **end if**
12: update ρ^- and $\rho^+;$
13: **until** stop criterion is met
14: **return** x^b

Figure 2. *Adaptive Large Neighborhood Search*

partly destroyed solution repairs it; that is, it returns a feasible solution built from the destroyed one. In line 2, the global best solution is initialized. In line 4, the weight *vectors* $r-$ and $r+$ are used to select the destroy and repair methods using a *roulette wheel principle.* The algorithm calculates the probability Φ_j of choosing the j^{th} destroy method as follows:

$$\phi_j^i = \frac{\rho_j^-}{\sum_{k=1}^{|\Omega^-|} \rho_k^-}$$

and the probabilities for choosing the repair methods are determined in the same way. In the line 5, the heuristic first applies the destroy method and then the repair method to obtain a new solution x_t. The new solution is evaluated, and the heuristic determines whether this solution should become the new current solution (line 6) or whether it should be rejected. The accept function can be implemented in different ways. The simplest choice is to only accept improving solutions. Line 9 checks whether the new solution is better than the best known solution. Here c(x) denotes the objective value of solution x. The best solution is updated in line 10 if necessary.

The weights are adjusted dynamically, based on the recorded performance of each destroy and repair method. This takes place in line 12: when an iteration of the ALNS heuristic is completed, a score y for destroy and repair method used in the iteration is computed using the formula:

$$\Psi = \begin{cases} \omega_1 & \text{if the new solution is a new global best} \\ \omega_2 & \text{if the new solution is better than the current one} \\ \omega_3 & \text{if the new solution is accepted} \\ \omega_4 & \text{if the new solution is rejected} \end{cases}$$

where *w1*, *w2*, *w3* and *w4* are parameters. A high value corresponds to a successful method. We would normally have w1 ≥w2 ≥w3 ≥w4 ≥ 0.

It is up to the implementer to choose the termination criterion, but a limit on the number of iterations or a time limit would be typical choices. In line 13, the termination condition is checked. In line 14, the best solution found is

Journal of Technical Science & Technologies; ISSN: 2298-0032; e-ISSN: 2346-8270; Volume 5, Issue 2, 2016

returned. From the pseudo code it can be noticed that the LNS meta-heuristic does not search the entire neighborhood of a solution, but merely samples this neighborhood.

A number of criteria can be used to measure how much a neighborhood contributes to the solution process: new best solutions are obviously given a large score, but also not previously visited solutions are given a score. Depending on the local search framework used on the master level, one may also give specific scores to accepted solutions, e.g. in a simulated annealing framework. Since each step of the ALNS heuristic involves two neighborhoods (destroy and a repair neighborhood), the score obtained in a given iteration is divided equally between them.

Methodology

The theoretical approach (finding the solution of the first stage: an initial (comprehensive) routing for all depot and all vehicles) described above, is implemented with the open-source toolkit Jsprit. Jsprit - a Java based, open source toolkit for solving rich traveling salesman (TSP) and vehicle routing problems (VRP))

Jsprit can solve problems with pickups and deliveries, back hauls, heterogeneous fleets, finite and infinite fleets, multiple depots, time windows, open routes, different start and end locations, and initial loads. Jsprit allows to define services (one stop) and shipments (two stops) with multiple capacity dimensions. Additionally, you can set time windows and required skills.

Jsprit allows us to add multiple depots by just adding vehicles/drivers with different start locations. One can specify start and end locations explicitly. Additionally, one can define a heterogeneous fleet by assigning different vehicle types (with different capacities and transportation costs), skills and operation times to your vehicles.Then, one has to put it all together to define the problem, set the routing costs and specify whether you have a finite or infinite fleet.

Finally, one solves the problem by defining and running an algorithm. Here it comes out-of-the-box (that is, the solution will be selected automatically based on the problem description).

We have developed a modification of the algorithm described above (written in the Jsprit framework). Namely, our algorithm takes into account a probability of links' congestion, estimation of probability of their release of busy route sections. Our modification of the algorithm can plan routes for any starting and finishing nodes. The modified algorithm will be intensively used for the second phase of the proposed approach. Namely, the algorithm will be used for optimal planning of each vehicle (driver) under circumstances of congested nearest two-three links. Detailed description of the proposed approach will be submitted in our next paper.

Conclusion

For visual representation of above stated VRP, example was created. As shown in Fig. 3, there are 20 stops, all located in Tbilisi. These are destinations where fleet of 12 vehicles from 3 different depots must visit. Every stop has its time windows, which cannot be violated. Time windows are shown as start time and end time. Each stop has its service duration and also quantity of goods which must be delivered.

After implementing *Jsprit* and applying longitude and latitude parameters of stops on the map, compilation and optimization is made. As shown in Fig. 5 using above mentioned techniques and tools we have built optimal route solution for 20 customers with 3 depots. We placed customers in different places in Tbilisi. Depots are located also in Tbilisi.

	id	job-id	type	name	address	latitude	longitude	service-duration	start-time	end-time	quantity
1	Stop1		D	Gardenia S...	Nikoloz Khu...	41,731	44,832	00:09:00	08:30:00	17:00:00	4 521
2	Stop2		D	Lit Geo Invest	Tsotne Dadi...	41,719	44,803	00:13:00	08:00:00	17:00:00	3 567
3	Stop3		D	Dinamo Are...	Archil Kurdi...	41,724	44,788	00:09:00	09:30:00	17:00:00	2 891
4	Stop4		D	Arttime	Heroes Squ...	41,714	44,782	00:05:00	09:00:00	18:00:00	5 012
5	Stop5		D	Vake Park	Ilia Chavcha...	41,712	44,751	00:16:00	09:00:00	18:00:00	2 987
6	Stop6		D	Metro Statio...	Vazha Pash...	41,724	44,731	00:09:00	08:30:00	18:00:00	3 120
7	Stop7		D	Lisi Lounge	Beritashvili ...	41,741	44,736	00:17:00	09:00:00	17:00:00	3 690
8	Stop8		D	Georgian P...	Grigol Roba...	41,756	44,775	00:22:00	08:00:00	18:00:00	1 706
9	Stop9		D	Gudushauri...	Nodar Bakh...	41,779	44,773	00:11:00	08:00:00	17:00:00	2 881
10	Stop10		D	Football Field	Mirian Mepe...	41,788	44,75	00:15:00	08:00:00	17:00:00	2 948
11	Stop11		D	Health Hou...	Avlipi Zurab...	41,692	44,826	00:07:00	09:30:00	17:00:00	1 721
12	Stop12		D	Geobotanik	Ketevan De...	41,686	44,848	00:05:00	08:30:00	17:00:00	3 786
13	Stop13		D	Koniaki Sett...	Davit Saraji...	41,794	44,801	00:08:00	09:30:00	17:00:00	4 587
14	Stop14		D	Internationa...	Grigol Pera...	41,797	44,774	00:16:00	08:30:00	18:00:00	3 031
15	Stop15		D	Foodmart	Vakhtang G...	41,765	44,751	00:12:00	09:30:00	18:00:00	3 554
16	Stop16		D	AutoNewStyle	Trialeti strt	41,694	44,877	00:08:00	09:30:00	18:00:00	3 417
17	Stop17		D	Rikhe	Wine Rise	41,692	44,811	00:08:00	08:30:00	17:00:00	3 662
18	Stop18		D	Tbilisi Airport	Europe Strt	41,672	44,96	00:13:00	08:30:00	18:00:00	1 729
19	Stop19		D	Judo Nation...	Akaki Belias...	41,761	44,78	00:08:00	09:30:00	18:00:00	4 475
20	Stop20		D	Embassy of...	Krtsanisi strt	41,675	44,82	00:05:00	08:00:00	17:00:00	6 213

Figure 3. *Customer location database*

All customer locations and depots are indicated on the map via longitude and latitude parameters. The geocoding is done using a database file that can couple zip codes to coordinates (Fig. 3). Routes of vehicles from depots to destinations are marked with different colors. While optimizing routes cost of fuel, total route l engths time windows and vehicle loads are considered. Traveled kilometers are calculated with OpenStreetmap (or any service that has geocoding and information about roads in Georgia like Google maps or Bing maps).

To visualize the process of mapping and routing, we need to review one route. As we have seen in Fig. 4, we

Journal of Technical Science & Technologies; ISSN: 2298-0032; e-ISSN: 2346-8270; Volume 5, Issue 2, 2016

have 3 depots with total fleet of vehicles – 12. For example, we take vehicle #1 from second depot. Second depot is located in Tbilisi (adr. Dadiani strt.)

Figure 4. *Multiple routing visualization*

As the parameters in this example case are set by us, we consider that depot 1 opens at 8:30 and first vehicle which is unoccupied at this moment has the capacity of 6500 boxes of cargo. Nearest stops, in our case customer's office demands, are generated randomly. Customer demands are: how much cargo they need daily (boxes) and also working hours of each office. Working hours in our case are hard Time Windows (TW). After randomizing parameters (quantity of boxes, working day start and end time and approximate service duration), we receive that nearest stops are in Vake (adr. Chavchavadze str.) and Vaja Pshavela (adr. V.Pshavela str.). After calculating possible variants of routes, we receive that optimal solution will be to load the whole cargo to the vehicle 1 with max load of 6500 at once and then go first to Vaja Pshavela with demand of 3120 boxes after that to Vake with demand of 2987 and after that back to the depot. Load of vehicles is shown in Figure 6.

Above described case is an example of methodology to get optimal routing and planning with simple demands and small amount of stops. With such planning we can calculate even bigger amount of stops and big amount of restrictions

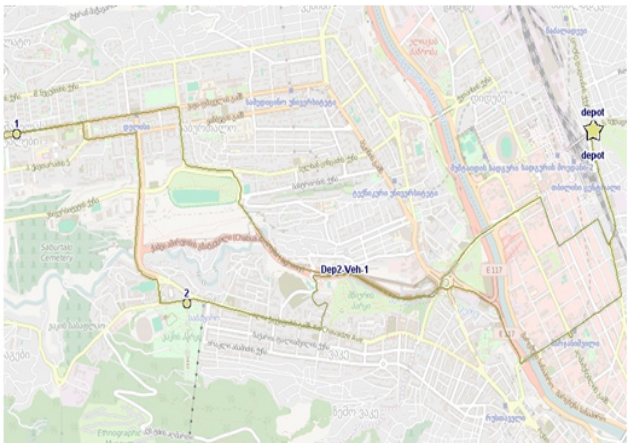

Figure 5. *Individual routing visualization*

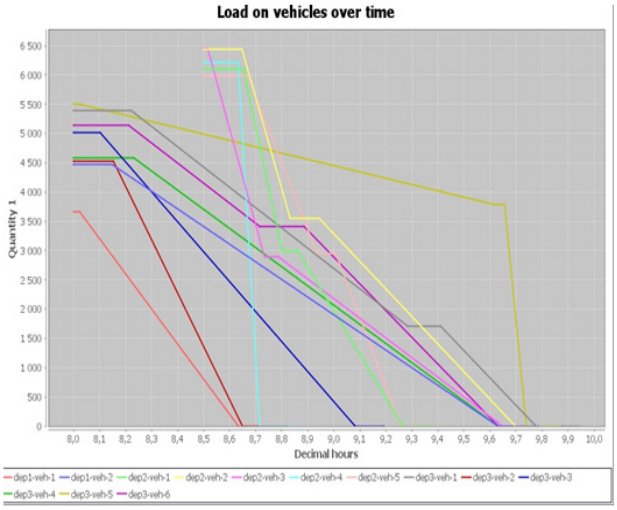

Figure 6. *Load of vehicles over time*

References

Guner, A.R., Alper, M., & Chinnam, R. B. (2012). Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information. *Computers & Operations Research* 39, 358–373.

Jsprit - a Java based, open source toolkit for solving rich traveling salesman (TSP) and vehicle routing problems (VRP), from http://jsprit.github.io/

Kok, A.L., Hans, E.W., & Schutten, J.M.J.Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Operational Methods for Production and Logistics, University of Twente, P.O. Box 217, 7500AE, Enschede, Netherlands,2013.

Lutz, R. (2014). Adaptive Large Neighborhood Search. Bachelor thesis at Ulm University

Pillac, V., Gueret, Ch., & Medaglia, A. (2011). Dynamic Vehicle Routing Problems: State of the art and Prospects. Technical Report 10/4/AUTO.. HAL Id: hal-00623474, https://hal.archives-ouvertes.fr/hal-00623474,2011

Pisinger. D., & Stefan, R. (2005). A general heuristic for vehicle routing problems, Computers and Operations Research, 176-193

Pisinger, D., & Stefan, R. (2004). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Submitted to Transportation Science.

Pisinger. D., & Ropke. S. (2010). Large neighborhood search. *Handbook of Metaheuristics.* (2 ed., pp. 399-420)., Springer.