Mixed Integer Programming Model For The Vehicle Routing Problem With Time Windows Considering Avoiding Route

Hotman Simbolon, Herman Mawengkang

Abstract— **The vehicle routing problem (VRP) is a combinatorial optimization problem to specify a homogeneous set of vehicles and routes, in which each vehicle starts from a depot and traverses along a route in order to serve a set of customers with known geographical locations, and then finish its tour at the same depot. The objective is to find the minimum total travel cost (related to the travel times or distances) and operational cost (related to the number of vehicles used). In this paper we present a variant of the VRP with time windows given a set of avoiding paths and to find the minimum total travel cost. We use integer programming model to describe the problem. A feasible neighbourhood approach is proposed to solve the model.**

Index Tems—**Vehicle routing problem with time windows, avoiding path, integer programming, feasible neighbourhood search**

I. INTRODUCTION

In the distribution of goods or services, there exists a problem that is known as the vehicle routing problem. In fact there is a wide variety of situationsand therefore the problem is not unique but a vast class of problems, each one with its own characteristics and constraints [5]. Vehicle Routing Problem is one of the important issues that exist in transportation system.

In the classical way the Vehicle Routing Problem (VRP) is defined as a problem of finding the optimal routes of delivery or collection from one or several depots to a number of cities or customers, while satisfying some constraints. Collection of household waste, gasoline delivery trucks, goods distribution, snowplough and mail delivery are the most used applications of the VRP. The VRP plays a vital role in distribution and logistics. Huge research efforts have been devoted to studying the VRP since 1959 where Dantzig and Ramser have described the problem as a generalised problem of Travelling Salesman Problem (TPS). Thousands of papers have been written on several VRP variants such as Vehicle Routing Problem with Time Windows (VRPTW), Vehicle Routing Problem with Pick-Up and Delivery (VRPPD) and Capacitated Vehicle Routing Problem (CVRP) [9]. However, many gaps may exist between the basic VRP and real application; for example, the number of depots, customer requirements with multiple pickups, delivery, type of vehicles with different travel times, travel costs and capacity, time windows restrictions, and route restrictions for vehicles. [10] discuss comprehensive details on VRPs consisting of its variants, formulation, and solution methods. [7] presents a compact review of vehicle routing literature.

Manuscript received August, 2014.

Hotman Simbolon, *University of HKBP Nomensen*
Herman Mawengkang*, Mathematics, University of Sumatera Utara, Medan, Indonesia,*

The Vehicle Routing Problem with Time Window (VRPTW) is a generalisation of the well-known VRP. It can be reviewed as a combined vehicle routing and scheduling problem which often arises in many real-world applications. It is to optimise the use of a fleet of vehicles that must make a number of stops to serve a set of customers, and to specify which customers should be served by each vehicle and in what order to minimise the cost, subject to vehicle capacity and service time restrictions [4]. The problem involves assignment of vehicles to trips such that the assignment cost and the corresponding routing cost are minimal.

In terms of graph the VRPTW can be defined as follows: Let $G = (V, A)$ be a connected digraph consisting of a set of n + 1 nodes, each of which can be reached only within a specified time interval or time window, and a set A of arcs with non-negative weights representing travel distances and associated travel times. Let one of the nodes be designated as the depot. Each node i, except the depot, requests a service of size q_i.

This paper discusses a variant of VRPTW in which there may be avoiding route. The avoiding sub-route involving pairs of edges occur frequently and can occur dynamically due to rush hour, lane closures, construction, etc. Longer forbidden subpaths are less common, but can arise, for example if heavy traffic makes it impossible to turn left soon after entering a multi-lane roadway from the right. If we are routing a single vehicle it is more natural to find a detour from the point of failure when a forbidden path is discovered.

Logically, a model whereby an algorithm identifies a potential path, and then this path is tried out on the actual network. In case of failure, further tests can be done to pinpoint a minimal forbidden subpath. Because such tests are expensive, a routing algorithm should try out as few paths as possible. In particular it is practically impossible to identify all forbidden paths ahead of time—we have an exponential number of possible paths to examine in the network. Therefore we impose an assumption of having no a priori knowledge of the forbidden paths, and of identifying forbidden paths only by testing feasibility of a path.

The problem can be defined as follows : given a graph $G(V, A)$, and vertices s and t, and given a set X of forbidden paths in G, find optimal set of route P such that no path in X is a subpath of P. Routes in *X are called* exceptions, and the desired route is called a shortest exception avoiding route. We allow an exception avoiding route to be non-simple graph, i.e., to repeat vertices and edges. In fact the problem becomes hard if the solution is restricted to simple [11]. This problem has been called the Shortest Path Problem with Forbidden Paths [12, 13].

II. PRELIMINARY

We are given a directed graph $G(V, A)$ with $n = |V|$ vertices and $m = |A|$ edges where each edge $e \in A$ has a positive weight denoting its length. We are also given a source vertex $s \in V$, a destination vertex $t \in V$, and a set X of route in G. The graph G together with X models a vehicle routing network in which a vehicle cannot follow any route in X because of the physical constraints .. We want to find a shortest route from s to t that does not contain any route in X as a subpath—we make the goal more precise as follows. A route is a sequence of vertices each joined by an edge to the next vertex in the sequence. Note that we allow a route to visit vertices and edges more than once. If a route does not visit any vertex more than once, we explicitly call it a simple route. *A* simple directed route from vertex v to vertex w in G is called a forbidden route or an exception if a vehicle cannot follow the route from v to w because of the physical constraints. Given a set X of forbidden route, a route $(v_1, v_2, v_3, \ldots, v_l)$ is said to avoid A if $(v_i, v_i + 1, \ldots, v_j) \notin A$ for all i, j such that $1 \leq i < j \leq l$. A route P from s to t is called a shortest A-avoiding route if the length of P is the shortest among all A-avoiding route from s to t. We will use the term "exception avoiding" instead of "X-avoiding" when A is equal to X, the set of all forbidden paths in G.

III. VEHICLE ROUTING PROBLEM

In the late fifties, [2] introduced the VRP, which can be viewed as an m-TSP with customer demands and vehicle capacity. An example of such a VRP is shown in Figure 1. The VRP introduced [2], strictly speaking, is called *Capacitated Vehicle Routing Problem* (CVRP), and is one of the simplest vehicle routing problems.

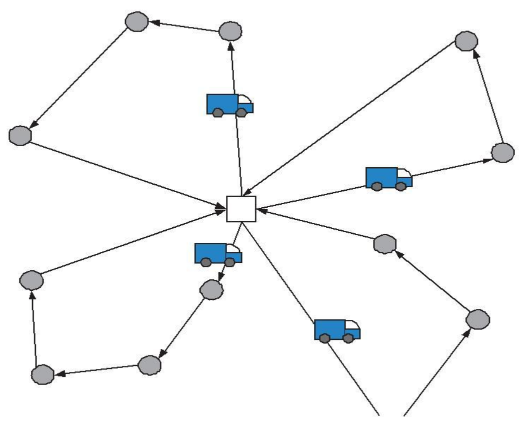

Figure 1: An example of the VRP.

III.1 Problem Formulation of VRPTW

Given a graph $G(V, A)$ with nodes $V = C \cup \{0\}$ and arcs A, where C is the set of customers, and 0 is the depot. Moreover, we have a set R of resources which e.g. can be load and/or time. Each resource $r \in R$ has a resource window $[\, a_i^r, b_i^r \,]$ that must be met upon arrival to node $i \in V$, and a consumption $t_{ij}^r \geq 0$ for using arc $(i, j) \in A$. A resource consumption at a node $i \in C$ is modeled by a resource consumption at edge (i, j), and hence usually $t_{0j}^r \geq 0$ for all $j \in C$. A global capacity limit Q can be modeled by imposing a resource window $[0, Q]$ for the depot node 0.

The VRP can now be stated as: Find a set of routes starting and ending at the depot node 0 satisfying all resource windows, such that the cost is minimized and all customers C are visited.

A solution to the VRP will consist of a number of routes

$$
\begin{aligned}
&0 \to i_1^1 \to \cdots \to i_{k_1}^1 \to 0,\\
&0 \to i_1^2 \to \cdots \to i_{k_2}^2 \to 0,\\
&\vdots\\
&0 \to i_1^n \to \cdots \to i_{k_n}^n \to 0
\end{aligned}
$$

where n is the number of vehicles, and k_j is the length of the j'th route.

III.2 Model

In the following let c_{ij} be the cost of arc $(i, j) \in A$, x_{ij} be the binary variable indicating the use of arc (i, j) $\in$ A, and t_{ij}^r (the resource stamp) be the consumption of resource r $\in$ R at the beginning of arc $(i, j) \in A$. Let $\delta^+(i)$ and $\delta^-(i)$ be the set of outgoing respectively ingoing arcs of node $i \in V$. Combining the two index model from Bard et al. [3] with the constraints ensuring the time windows for the TSP by Ascheuer et al. [1] a mathematical model can be formulated as follows:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{(s,j)\in\delta^+(i)} x_{ij} = 1, \ \forall\, i \in C, \tag{2}$$

$$\sum_{(s,j)\in\delta^-(i)} x_{ij} = \sum_{(s,j)\in\delta^+(i)} x_{ij}, \ \forall\, i \in V \tag{3}$$

$$\sum_{(s,j)\in\delta^-(i)} (T_{ji}^r + t_{ji}^r x_{ij}) \leq \sum_{(s,j)\in\delta^+(i)} T_{ji}^r, \ \forall\, r \in R, \ \forall\, i \in C, \tag{4}$$

$$a_i^r x_{ij} \leq T_{ji}^r \leq b_i^r x_{ij} \quad \forall\, r \in R, \ \forall\, (i, j) \in A, \tag{5}$$

$$T_{ij}^r \geq 0, \ \forall\, r \in R, \ \forall\, (i, j) \in A, \tag{6}$$

$$x_{ij} \in \{0,1\}, \ \forall\, (i, j) \in A, \tag{7}$$

The objective (1) sums up the cost of the used arcs. Constraints (2) ensure that each customer is visited exactly once, and (3) are the flow conservation constraints. Constraints (4) and (5) ensure the resource windows are satisfied. It is assumed that the bounds on the depot are always satisfied. Note, that no sub-tours can be present since only one resource stamp per arc exists and the arc weights are positive for all $(i, j) \in A : i \in C$.

For a one dimensional resource such as *load* a stronger lower bound of the linear programming relaxation can be obtained by replacing (4) to (6) with $\Sigma_{(i,j)\in\delta^+(S)} x_{ij} \geq r(S)$, where $r(S)$ is a minimum number of vehicles needed to service the set S. All though this model can not be directly solved it is possible to overcome this problem by only including the constraints that are violated. For more details on how to separate the constraint and calculate the value of $r(S)$ the reader is refered to [10].

III.3 Model for VRPTW with avoiding route

Given a set X of avoiding route, a route $(v_1, v_2, v_3, \ldots, v_l)$ is said to avoid A if $(v_i, v_i + 1, \ldots, v_j) \notin A$ for all i, j such that $1 \leq i < j \leq l$. A route P from s to t is called a shortest A-avoiding route if the length of P is the shortest among all A-avoiding route from s to t. We will use the term "exception avoiding" instead of "X-avoiding" when A is equal to X, the set of all forbidden paths in G. It is necessarily to assume that costumers are not in the specified route. The mixed integer programming model can be written as :

$$\min \sum_{(i,j)\in A;(i,j)\notin X} c_{ij}x_{ij} \tag{8}$$

$$\text{s.t.} \sum_{(s,j)\in\delta^+(i)} x_{ij} = 1,\ \forall\, i \in C,\, i \notin X, \tag{9}$$

$$\sum_{(s,j)\in\delta^-(i)} x_{ij} = \sum_{(s,j)\in\delta^+(i)} x_{ij},\ i \notin X,\ \forall\, i \in V, \tag{10}$$

$$\sum_{(s,j)\in\delta^-(i)} (T^r_{ji} + t^r_{ji}x_{ij}) \leq \sum_{(s,j)\in\delta^+(i)} T^r_{ji},$$
$$i \notin X,\ \forall\, r \in R,\ \forall\, i \in C, \tag{11}$$

$$a^r_i x_{ij} \leq T^r_{ji} \leq b^r_i x_{ij},\ (i,j) \notin X, \forall\, r \in R,\ \forall\,(i,j) \in A, \tag{12}$$

$$T^r_{ij} \geq 0,\ (i,j) \notin X, \forall\, r \in R,\ \forall\,(i,j) \in A, \tag{13}$$

$$x_{ij} \in \{0,1\},\ (i,j) \notin X, \forall\,(i,j) \in A, \tag{14}$$

In the above model every vehicle assigned will not be travelling along the avoiding route.

IV. NEIGHBOURHOOD SEARCH

It should be noted that, generally, in integer programming the reduced gradient vector, which is normally used to detect an optimality condition, is not available, even though the problems are convex. Thus we need to impose a certain condition for the local testing search procedure in order to assure that we have obtained the "best" suboptimal integer feasible solution.

Further in [3] has proposed a quantity test to replace the pricing test for optimality in the integer programming problem. The test is conducted by a search through the neighbours of a proposed feasible point to see whether a nearby point is also feasible and yields an improvement to the objective function.

Let $[\beta]_k$ be an integer point belongs to a finite set of neighbourhood $N([\beta]_k)$ We define a neighbourhood system associated with $[\beta]_k$, that is, if such an integer point satisfies the following two requirements

1. if $[\beta]_j \in N([\beta]_k)$ then $[\beta]_k \in [\beta]_j$, j $\neq$ k.
2. $N([\beta]_k) = [\beta]_k + N(0)$

With respect to the neighbourhood system mentioned above, the proposed integerizing strategy can be described as follows.

Given a non-integer component, x_k, of an optimal vector, x_B. The adjacent points of x_k, being considered are $[x_k]$ dan $[x_k] + 1$. If one of these points satisfies the constraints and yields a minimum deterioration of the optimal objective value we move to another component, if not we have integer-feasible solution.

Let $[x_k]$ be the integer feasible point which satisfies the above conditions. We could then say if $[x_k] + 1 \in N([x_k])$ implies that the point $[x_k] + 1$ is either infeasible or yields an inferior value to the objective function obtained with respect to $[x_k]$. In this case $[x_k]$ is said to be an "optimal" integer feasible solution to the integer programming problem. Obviously, in our case, a neigbourhood search is conducted through proposed feasible points such that the integer feasible solution would be at the least distance from the optimal continuous solution.

V. THE ALGORITHM

First we solve the mixed integer programming model Eq. (8) – (14) be relaxing the integer restriction. After solving the relaxed problem, the procedure for searching a suboptimal but integer-feasible solution from an optimal continuous solution can be described as follows.
Let

$$x = [x] + f, \quad 0 \leq f \leq 1$$

be the (continuous) solution of the relaxed problem, $[x]$ is the integer component of non-integer variable x and f is the fractional component.

Stage 1.

Step 1. Get row i^* the smallest integer infeasibility, such that $\delta_{i^*} = \min\{f_i, 1 - f_i\}$
(This step is taken in order to get a minimum deterioration of the objective function).

Step 2. Do a pricing operation
$v^T_{i^*} = e^T_{i^*}B^{-1}$

Step 3. Calculate $\sigma_{ij} = v^T_{i^*}\alpha_j$
With j corresponds to

$$\min_j \left\{ \left| \frac{d_j}{\alpha_{ij}} \right| \right\}$$

Calculate the maximum movement of nonbasic j at lower bound and upper bound.
Otherwise go to next non-integer nonbasic or superbasic j (if available). Eventually the column j^* is to be increased form LB or decreased from UB. If none go to next i^*.

Step 4. Solve $B\alpha_{j^*} = \alpha_{j^*}$ for α_{j^*}

Step 5. Do ratio test for the basic variables in order to stay feasible due to the releasing of nonbasic j^* from its bounds.

Step 6. Exchange basis

Step 7. If row $i^* = \{\emptyset\}$ go to Stage 2, otherwise
Repeat from step 1.

Stage 2. Pass1 : adjust integer infeasible superbasics by fractional steps to reach complete integer feasibility.
Pass2 : adjust integer feasible superbasics. The objective of this phase is to conduct a highly localized neighbourhood search to verify local optimality.

VI. CONCLUSIONS

This paper presents a VRPTW model in which there are some avoiding route. The framework of the model stems from VRP with time windows. Then exclude the avoiding route from the previous assigned route. We solve the model using a feasible neighbourhood search.

REFERENCES

[1] Daniel Villeneuve and Guy Desaulniers, “The shortest paths with forbidden paths”, European Journal of Operational Research, Vol. 165, No. 1, pp. 97-107, 2005.

[2] G.B. Dantzig and J.H. Ramser, “The truck dispatching problem”, Management Science, Vol. 6, 80, 1959.

[3] H. E Scarf., *Testing for optimality in the absence of convexity* in Walter P Heller, Ross M Starr and David A Starrett (Eds) (Cambridge University Press) pp 117-134, 1986.

[4] I. Ellabib , A. B. Otman, & P. Calamai. “An Experimental Study of a Simple Ant Colony System for the Vehicle Routing Problem with Time Windows”. ANTS, LNCS 2463: 53-64, 2002.

[5] L. Berbotto, S. Garcia, F. J. Nogales., “A Vehicle Routing Model with Split Delivery and Stop Nodes”. Working Paper 11-09(06), Statistics and Econometric Series, Universidad Carlos III de Madrid, 2011.

[6] M. Ahmed and A. Lubiw, “Shortest path avoiding forbidden subpaths”, Symposium on Theoretical Aspect of Computer Science, pp. 63-74, 2009.

[7] M. Drexl.” *Rich Vehicle Routing in Theory and Practice”*. Technical Report LM-2011-04, Johannes Gutenberg University Mainz, 2011.

[8] N. Ascheuer, M. Fischetti, and M. Grätschel, “Solving the asymmetric travelling salesman problem with time windows by branch-and-cut”, Mathematical Programming, Vol. 90, No. 3, pp. 475-506, 2001.

[9] N. Christofides, A. Mingozzi, and P. Toth., "The vehicle routing problem", In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C., (Eds.), *Combinatorial optimization*. Chicester: John Wiley, pp. 315–338, 1979.

[10] P. Toth and D. Vigo, “An overview of vehicle routing problems”, In P. Toth and D. Vigo, Editors, The Vehicle Routing Problem, Chapter 1, pp. 1-26, SIAM, 2002.

[11] Stefan Szeider, “Finding paths in graphs avoiding forbidden transitions”, Discrete Appl. Math., Vol. 126, No. 2-3, pp. 261-273, 2003.

[12] Daniel Villeneuve and Guy Desaulniers, The shortest paths with forbidden paths, European *Journal of Operational Research, Vol. 165, No. 1*, 2005, pp. 97-107.

[13] M. Ahmed and A. Lubiw, Shortest path avoiding forbidden subpaths, *Symposium on Theoretical Aspect of Computer Science,* 2009, pp. 63-74.

Hotman Simbolon Earned a PhD degree from the Graduate School of Mathematics, University of Sumatera Utara. He received his Masters from Institute Pertanian Bogor, Indonesia. He is a senior staff in the department of Mathematics, University of HKBP Nommensen Siantar, Indonesia.

Herman Mawengkang, He is a professor of Operations Research at the University of Sumatera Utara. He gained his PhD from the University of New South Wales, Sydney, Australia. He already produced many journal papers published internationally. Currently he is the chairman of Graduate School of Mathematics, University of Sumatera Utara.