



Received: 16 December 2015  
Accepted: 03 March 2016  
First Published: 09 March 2016

\*Corresponding author: Xin Chen,  
Department of Industrial Engineering,  
Southern Illinois University Edwardsville,  
Edwardsville, IL 62026-1805, USA  
E-mail: [xchen@siue.edu](mailto:xchen@siue.edu)

Reviewing editor:  
Ryan Loxton, Curtin University, Australia

Additional information is available at  
the end of the article

## APPLIED & INTERDISCIPLINARY MATHEMATICS | RESEARCH ARTICLE

# Development and implementation of algorithms for vehicle routing during a no-notice evacuation

Xin Chen<sup>1\*</sup>, Zhu Zhang<sup>1</sup> and Ryan Fries<sup>2</sup>

**Abstract:** This article develops and implements time-dependent shortest path and assignment algorithms for vehicle routing and assignment during a no-notice evacuation. A time-dependent shortest path algorithm with arc labeling is designed to improve computer storage space efficiency. Visual Basic for Application (VBA) is used to improve time efficiency of both shortest path and assignment algorithms. Performance of the algorithms is analyzed and compared through implementation in VBA and the General Algebraic Modeling System (GAMS). Since VBA seamlessly integrates with Microsoft Excel and enables efficient data manipulation, the algorithms implemented in VBA are more efficient and obtain optimal solutions faster than those implemented in the GAMS. The shortest path algorithm with arc labeling implemented in VBA may be used for real-time vehicle routing for large road networks during no-notice evacuations.

**Subjects:** Engineering & Technology; Industrial Engineering & Manufacturing; Operations Research; Technology

**Keywords:** arc labeling; assignment problem; evacuation; time-dependent shortest path; VBA; vehicle routing

### 1. Introduction

In a generalized vehicle routing problem (VRP), multiple vehicles travel from origins (depots; Montoya-Torres, Franco, Isaza, Jiménez, & Herazo-Padilla, 2015) to destinations (demand points). One of the objectives of the VRP is to minimize total travel time or the latest arrival time for a fleet of vehicles while meeting demand at destinations. The VRP has a wide range of applications,

### ABOUT THE AUTHORS

Xin Chen, PhD, is Associate Professor of Industrial Engineering at Southern Illinois University Edwardsville. His research focuses on network- and knowledge-centric collaborative control with applications in critical infrastructure protection and management, energy distribution, social networks, supply chains, and transportation. This article is part of his research on applying optimization tools in large, complex networks.

Zhu Zhang was a graduate student in Industrial Engineering at Southern Illinois University Edwardsville. He graduated in 2014 with a Master of Science in Industrial Engineering.

Ryan Fries is an associate professor in the Department of Civil Engineering at Southern Illinois University Edwardsville. His research focuses on design and planning of transportation systems.

### PUBLIC INTEREST STATEMENT

During emergencies, evacuations, and many other situations, it is important for emergency vehicles from various locations to reach their respective destinations quickly. This article develops an efficient algorithm that calculates the best routes for multiple vehicles. When road conditions change, the algorithm computes and updates the routes within a few minutes, and can be used for real-time route planning in large road networks.

e.g. bank or postal deliveries and school bus routing. During a no-notice evacuation of a metropolitan area, emergency response vehicles must be swiftly deployed to multiple demand points to manage traffic, including blocking ramps and lanes and diverting traffic. Once an emergency response vehicle arrives at an intended demand point, it is stationed at the point until the evacuation completes, which may take from a few hours to one day. Certain demand points may need multiple emergency response vehicles whereas others may need only one vehicle. Before a no-notice evacuation begins, emergency response vehicles are usually stationed at depots controlled by state or local government agencies, e.g. the state department of transportation. Since road conditions change rapidly, there is a strong need for real-time emergency vehicle routing during a no-notice evacuation.

The objective of this article is to investigate effective algorithms which may be implemented to identify optimal vehicle assignments and routes for the VRP in real time. The main contributions of this article are as follows:

- (a) Implementing and integrating time-dependent shortest path (TDSP) and assignment algorithms to minimize total travel time for emergency response vehicles during a no-notice evacuation of a metropolitan area or other emergency response and evacuation situations; and
  - (b) Improving time and space efficiency of the algorithms for real-time vehicle assignment and routing in large road networks.
- (i) The TDSP algorithm with arc labeling requires less computer memory compared to node labeling, and improves space efficiency. The TDSP with arc labeling and assignment algorithms are applied to road networks with up to 2,000 nodes, and are able to identify the optimal vehicle routes and assignments within eight minutes. (The largest road network studied previously had 1,000 nodes; Almoustafa, Hanafi, & Mladenović, 2013); and
  - (ii) The algorithms are implemented in Visual Basic for Applications (VBA®). Input and output of the algorithms are stored and processed using Microsoft Excel®. The seamless integration of VBA® and Excel® greatly improves time efficiency and enables real-time vehicle assignment and routing.

## 2. Background

Research on the VRP stemmed from the traveling salesman problem (Dantzig, Fulkerson, & Johnson, 1954) and was further developed in some early applications (Balas & Toth, 1985; Clarke & Wright, 1964; Haimovich, Rinnooy Kan, & Stougie, 1988; Laporte, Nobert, & Taillefer, 1987). Recent research focused on heuristic and meta-heuristic approaches (e.g. Bräysy & Gendreau, 2005; Zare-Reisabadi & Mirmohammadi, 2015). Most early studies assumed vehicle travel time between two nodes was static and did not change regardless of traffic. During the last decade, time-dependent vehicle routing problems (TDVRPs) have gained increasing attention (Almoustafa et al., 2013; Ando & Taniguchi, 2006; Andres Figliozzi, 2012; Chen, Hsueh, & Chang, 2006; Chen et al., 2013; Haghani & Jung, 2005; Ichoua, Gendreau, & Potvin, 2003; Kritzing et al., 2012; Lecluyse, Van Woensel, & Peremans, 2009; Maden, Eglese, & Black, 2010; Spliet & Gabor, 2012; Vidal, Crainic, Gendreau, & Prins, 2014; Van Woensel, Kerbache, Peremans, & Vandaele, 2008). In the TDVRP, travel time between two directly connected nodes is dynamic and depends on many factors such as traffic. In many cases, travel time between two nodes may be described as a function of the node and time at which a vehicle begins to travel and the node at which the vehicle plans to arrive.

The VRP has many input parameters, including the number and origins of vehicles, the size and structure of a road network, the number and location of destinations, and vehicle travel times (dynamic or static). Numerous methods were introduced in recent years to solve the VRP; these methods stipulated various conditions for one or more parameters. One of the most common conditions was the upper limit for the size of road networks. Other conditions include static travel time

and upper limit for the number of destinations. These methods become ineffective (solutions far from optimal) or inefficient (cannot identify a good or optimal solution within an acceptable amount of time) when stipulated conditions do not hold. To develop effective and efficient algorithms to solve VRPs with many parameter values remain a considerable challenge (Vidal et al., 2014).

Since exact methods that identify optimal vehicle assignments and routes are either ineffective or inefficient for generalized VRPs, heuristic methods including the genetic algorithm (Haghani & Jung, 2005), tabu search (Ichoua et al., 2003), branch and price (Almoustafa et al., 2013), and column generation algorithm (Spliet & Gabor, 2012) were studied. Haghani and Jung (2005) presented a genetic algorithm to solve a pick-up or delivery VRP with soft time windows. The study considered multiple vehicles with different capacities, real-time service requests, and dynamic travel times between destinations. Ichoua et al. (2003) conducted experiments to solve the VRP with time-dependent travel speeds, which satisfy the first-in-first-out (FIFO) property, using a parallel tabu search heuristic. Almoustafa et al. (2013) improved a branch-and-bound method to solve the asymmetric distance-constrained VRP suggested by Laporte et al. (1987). Chen et al. (2006) formulated a real-time TDVRP with time windows as a series of mixed integer programming models and developed a heuristic algorithm, which included route construction and improvement. Spliet and Gabor (2012) proposed a formulation of a time window asymmetric VRP and developed two variants of a column generation algorithm to solve the linear programming relaxation of this formulation. Kritzing et al. (2012) applied variable neighborhood search algorithm to solve the TDVRP with time windows. Maden et al. (2010) proposed a heuristic algorithm for the VRP to minimize total travel time.

Road networks with different sizes were also studied. Laporte, Nobert, and Taillefer (1988) examined a class of asymmetrical multi-depot VRPs and location-routing problems for a network of 80 nodes. Haghani and Jung (2005) solved the TDVRP for networks with 30 demand nodes over 30 time intervals. Almoustafa et al. (2013) solved an asymmetric distance-constrained VRP for a network of 1,000 demand nodes. In summary, previous research predominately focused on developing heuristic methods for subsets of VRPs or TDVRPs. Effective and efficient algorithms which may be applied to generalized TDVRPs to obtain optimal vehicle assignments and routes were not available. Most algorithms and methods developed in previous research were not tested using real-world road networks and could not be validated for effectiveness or efficiency. The objective of this article is to provide effective and efficient algorithms for generalized TDVRPs in a no-notice evacuation. The rest of this article is organized as follows: Section 3 presents the problem and methodology. Section 4 validates the algorithms using two real-world and five simulated road networks. Section 5 concludes the article with future research directions.

### 3. Methodology

#### 3.1. Problem definition

Let  $A$ ,  $B$ ,  $\Gamma$  represent sets of vehicles, depots, and demand points, respectively, in a time-dependent road network. There are total  $|A|$  vehicles stationed at  $|B|$  depots at the beginning of a planning period, which is a time period during which vehicles must be dispatched to demand points to meet the demand. Some or all of the  $|A|$  vehicles need to be dispatched to  $|\Gamma|$  demand points, each of which requires  $d_\gamma$  vehicles, where  $\gamma$  represents a demand point,  $\gamma \in \Gamma$ . Let  $c_{ij,t}^\alpha$  be the cost (time) it requires for a vehicle  $\alpha$ ,  $\alpha \in A$ , to travel from node  $i$  at time  $t$  and to node  $j$ ,  $i, j \in V$ , where  $V$  is the node set in the road network.  $B, \Gamma \subset V$ . Let  $(i, j)$  represent an arc that originates from node  $i$  and points at node  $j$ ,  $(i, j) \in E$ , where  $E$  is the set of arcs in the road network.  $c_{ij,t}^\alpha = \infty$  if  $(i, j) \notin E$ . When  $i = \beta$ ,  $\beta$  is a depot and  $\beta \in B$ ,  $c_{\beta j,t}^\alpha = \infty$ ,  $\forall j$ , if  $\alpha$  is not ready to travel from  $\beta$  at time  $t$ .

In the TDVRP, the first objective is to identify the earliest time for  $\alpha$  to arrive at a demand point  $\gamma$  from its depot  $\beta$ .  $\alpha$  may travel from  $\beta$  at any time after  $\alpha$  is ready for travel. Suppose  $\alpha$  may travel from  $\beta$  at time  $t_1$  and arrive at  $\gamma$  at time  $t_2$ . Alternatively,  $\alpha$  may travel from  $\beta$  at time  $t_3$  and arrive at  $\gamma$  at time  $t_4$ . According to the FIFO property,  $t_2 < t_4$  if  $t_1 < t_3$ . Therefore,  $\alpha$  should travel from  $\beta$  as soon as  $\alpha$  is ready for travel. Let  $t_\beta$  be the time at which  $\alpha$  becomes ready for travel at  $\beta$ .  $c_{\beta j,t}^\alpha = \infty$  when  $t < t_\beta$ .

$\forall j. c_{\beta,j,t}^\alpha \ll \infty$  when  $t \geq t_\beta$  and  $(\beta, j) \in E$ . The first objective of the TDVRP is therefore to identify the TDSP for  $\alpha$  to travel from  $\beta$  at  $t_\beta$  to  $\gamma$ . Equation (1) is a model whose optimal solution is the TDSP between  $\beta$  and  $\gamma$  when  $\alpha$  travels from  $\beta$  at  $t_\beta$ .

$$\text{Minimize } \sum_i \sum_j c_{i,j,t}^\alpha X_{i,j}$$

Subject to:

$$\sum_j X_{i,j} - \sum_j X_{j,i} = 1 \quad \text{when } i = \beta$$

$$\sum_j X_{i,j} - \sum_j X_{j,i} = -1 \quad \text{when } i = \gamma$$

$$\sum_j X_{i,j} - \sum_j X_{j,i} = 0 \quad \text{when } i \neq \beta \text{ and } i \neq \gamma$$

$$X_{i,j} = \begin{pmatrix} 1 & (i,j) \text{ is on the path for } \alpha \text{ to travel from } \beta \text{ to } \gamma \\ 0 & \text{otherwise} \end{pmatrix} \quad \forall i, j \in V \quad (1)$$

The second objective of the TDVRP is to minimize total travel time. Let  $s_{\beta,\gamma}^\alpha$  represent the optimal value of Equation (1), i.e. the minimum time for  $\alpha$  to travel from  $\beta$  and arrive at  $\gamma$ . Equation (2) models an assignment problem (AP) that determines which vehicles are dispatched to a demand point to meet the demand. Note that the objective of Equation (2) is not to minimize the summation of arrival times or the latest arrival time. Since Equation (1) identifies the earliest arrival times, Equation (2) intends to minimize total travel time. In many VRPs, less travel time implies less uncertainty and more reliable vehicle routing and assignment (Chen et al., 2013). Both Equations (1) and (2) are pure integer programming problems. If  $\sum_{\gamma=1}^{|\Gamma|} d_\gamma = |A|$ , Equation (2) is a balanced transportation problem and both constraints may be changed to equality constraints. If  $\sum_{\gamma=1}^{|\Gamma|} d_\gamma > |A|$ , Equation (2) is infeasible.

$$\text{Minimize } \sum_{\alpha=1}^{|A|} \sum_{\gamma=1}^{|\Gamma|} s_{\beta,\gamma}^\alpha Y_{\alpha,\gamma}$$

Subject to:

$$\sum_{\gamma=1}^{|\Gamma|} Y_{\alpha,\gamma} \leq 1, \forall \alpha$$

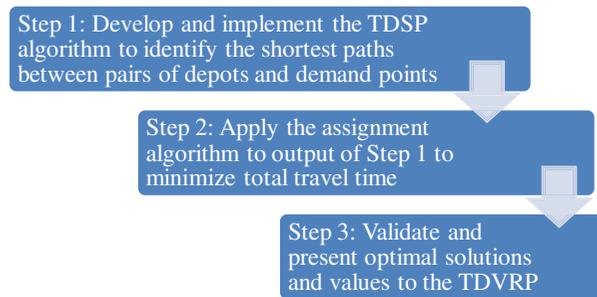
$$\sum_{\alpha=1}^{|A|} Y_{\alpha,\gamma} \geq d_\gamma, \forall \gamma$$

$$Y_{\alpha,\gamma} = \begin{pmatrix} 1 & \alpha \text{ travels to } \gamma \\ 0 & \text{otherwise} \end{pmatrix} \quad (2)$$

### 3.2. Solution steps

Optimization software packages may be used to solve models in Equations (1) and (2). For example, the General Algebraic Modeling System (GAMS; GAMS Development Corporation, 2013) is a high-level modeling system for mathematical programming and optimization. Equations (1) and (2) may be described in algebraic statements in GAMS input files and solvers may be used to find optimal solutions and values. For medium to large road networks, however, this approach is ineffective or inefficient due to the limitation of computer memory and the inconvenience of extremely long

**Figure 1. Three steps to solve the TDVRP for real-time transportation planning.**



computation time. For real-time transportation planning, time- and space-efficient algorithms must be developed to solve the models in Equations (1) and (2). Figure 1 shows three steps to identify optimal solutions and values to the TDVRP.

### 3.3. TDSP algorithms

Real-time TDVRPs remain a great challenge due to time and space complexities. In the TDVRP, characteristics of the road network change with time; a shortest path computed from a snapshot of the road network may not be the shortest path at a different time. The TDSP algorithm is developed to find the shortest paths between depots and demand points in real-time. Three assumptions related to the TDSP algorithm are:

- (a) The road network satisfies the FIFO principle, which specifies that if two vehicles take the same route from the same depot to the same demand point, the vehicle leaving the depot early always arrives at the demand point early. According to the FIFO principle, a vehicle should leave its depot or other nodes whenever it is ready. Waiting at any node is never beneficial because a vehicle that leaves later always arrives later;
- (b) The planning period is “discretized” into sufficiently small time intervals,  $\delta$ 's,  $\delta = 1, 2, 3, \dots$  and  $\delta \in \Delta$  where  $\Delta$  is the set of time intervals over the planning period; and
- (c) Travel time between two nodes connected by an arc depends on the time at which a vehicle leaves the beginning node of the arc.

The TDSP algorithm with node labeling was first developed by Dreyfus (1969) and further validated by Kaufman and Smith (1993). The TDSP algorithm with node labeling listed below finds the earliest arrival time of a vehicle at a demand point given that the vehicle is stationed at a depot when travel begins. The algorithm is executed for each pair of demand point and depot for each vehicle. Vehicles stationed at the same depot may be ready to travel from the depot at different times because different vehicles may require different preparation times, e.g. time for a driver to arrive at the depot and get ready may vary. The earliest arrival times of vehicles, which are stationed at the same depot, at the same demand point may be different and need to be calculated separately using the TDSP algorithm.

#### 3.3.1. TDSP algorithm with node labeling

- (1) Assign to every node  $i, i \in V$ , in a transportation network a value representing the arrival time of a vehicle  $\alpha, \alpha \in A$ , at the node. For a depot node  $\beta$  where  $\alpha$  is stationed, set the value to a finite positive integer number representing the time at which  $\alpha$  is ready to travel from  $\beta$ . Set the value to infinity for all other nodes  $i$ 's,  $i \in V$  and  $i \neq \beta$ ;
- (2) Mark  $\beta$  visited. Mark all other  $i$ 's unvisited. Set  $\beta$  as the current node;
- (3) Calculate the arrival time of  $\alpha$  at each unvisited neighbor  $j, j \in V$ , of the current node  $i, i \in V$ . A node  $j$  is a neighbor of  $i$  if there is an arc begins at  $i$  and points at  $j$ , i.e.  $(i, j) \in E$ . The arrival

time at  $j$  is the summation of the value set for  $i$  and the travel time  $C_{ij,\delta}^\alpha$  between  $i$  and  $j$ . The travel time  $C_{ij,\delta}^\alpha$  is obtained from a three-dimensional matrix,  $|V| \times |V| \times |\Delta|$ , which stores time-dependent travel times. For example, if  $i = 1, j = 2$ , and the value set for node 1 is 15, the travel time between nodes 1 and 2,  $C_{1,2,15}^\alpha$  is a component in the matrix identified by node 1, node 2, and  $\delta = 15$ .  $\delta = 15$  indicates  $\alpha$  begins traveling from node 1 at time 15;

- (4) For each unvisited neighbor  $j$  of the current node  $i$ , compare the arrival time at  $j$  calculated in Step 3 and the value set for  $j$ . Set the value for  $j$  as the smaller one between the two;
- (5) Identify the unvisited node with the smallest value. Mark the node visited. Set the node as the current node. If the node is the desired demand point  $\gamma, \gamma \in \Gamma$ , stop. The value set for  $\gamma$  is the earliest arrival time of  $\alpha$  traveling from  $\beta$  at  $\gamma$ . Otherwise go to Step 3.

The TDSP algorithm with node labeling is implemented in VBA® and the GAMS. The algorithm can only solve TDVRPs for small road networks with less than 500 nodes when there are more than 1,000 time intervals using a Windows 7 ×64 PC, Intel i7-3770 CPU @3.40 GHZ, and 16.0 GB RAM. The main reason for size limitation on the road network is the large storage space required by node labeling. The algorithm needs to manipulate a three-dimensional matrix,  $|V| \times |V| \times |\Delta|$ , for node labeling. Each component in the matrix is travel time from one node to the other during a time interval. These travel times are often obtained through field observations (Rakha, El-Shawarby, Arafeh, & Dion, 2006) and real-time monitoring of road network conditions.  $|V|$  is the size of the road network, i.e. the number of nodes.  $|\Delta|$  is the number of time intervals. For example, if the road network size increases by tenfold, the storage space for the three-dimensional matrix increases by 100 times.

The degree of a node in a network is the number of arcs connected to the node. Most real-world road networks have a mean degree between two and four (Barabasi, 2002; Jeong, 2003). On average, each node is connected to two to four arcs. If arc labeling is used, the TDSP algorithm manipulates a two-dimensional matrix,  $|E| \times |\Delta|$ , where  $|E|$  is the number of arcs in the road network. Each component in the two-dimensional matrix is travel time along an arc during a time interval. Travel times in the two-dimensional matrix are the same as those in the three-dimensional matrix, but are organized in a different format that reduces storage space requirement. Since  $|E| \leq 4 \times |V|$  according to the mean degree of a road network, storage space requirement for arc labeling is at most  $4 \times |V| \times |\Delta|$ , which is much less than  $|V| \times |V| \times |\Delta|$  required for node labeling for large road networks. The TDSP algorithm with arc labeling described below is used to identify the earliest arrival times of vehicles at demand points. This is a significant improvement for the TDSP algorithm and greatly increases space efficiency of the algorithm.

### 3.3.2. TDSP algorithm with arc labeling

- (1) Assign to every arc  $(i, j), (i, j) \in E$ , in a transportation network a value representing the arrival time of a vehicle  $\alpha, \alpha \in A$ , at  $j, i, j \in V$ . For  $(i, j)$  in which  $i = \beta$ , a depot node where  $\alpha$  is stationed, set the value to a finite positive integer number representing the time at which  $\alpha$  arrives at  $j$ . The arrival time at  $j$  is the summation of time at which  $\alpha$  is ready to travel from  $\beta$  and travel time from  $\beta$  to  $j$ . The travel time from  $\beta$  to  $j, C_{\beta j, \delta}^\alpha$  is obtained from a two-dimensional matrix,  $|E| \times |\Delta|$ , which stores time-dependent travel times. For example, if  $\beta = 1, j = 2$ , and the time at which  $\alpha$  is ready to travel from  $\beta$  is  $\delta = 15$ , the travel time between nodes 1 and 2,  $C_{1,2,15}^\alpha$  is a component in the matrix identified by arc  $(1, 2)$  and  $\delta = 15$ . Set the value to infinity for all other arcs  $(i, j), (i, j) \in E$  and  $i \neq \beta$ ;
- (2) Mark all  $(i, j)$  unvisited;
- (3) Identify the unvisited  $(i, j)$  with the smallest value. Mark  $(i, j)$  visited. Set the destination node, i.e. the second node  $j$ , in  $(i, j)$  as the current node. If  $j$  is the desired demand point  $\gamma, \gamma \in \Gamma$ , stop. The value set for  $(i, \gamma)$  is the earliest arrival time of  $\alpha$  traveling from  $\beta$  at  $\gamma$ ;
- (4) For each unvisited  $(i, j)$  whose  $i$  is the current node, compare the arrival time at  $j$  and the value set for  $(i, j)$ . The arrival time at  $j$  is the summation of time at which  $\alpha$  arrives at  $i$  and travel time from  $i$  to  $j$ . The arrival time at  $i$  is the value set for the arc marked as visited in Step 3. The travel

time from  $i$  to  $j$ ,  $C_{ij,\delta}^a$ , is obtained from the matrix  $|E| \times |\Delta|$ .  $\delta$  is the value set for the arc marked as visited in Step 3;

- (5) For each unvisited  $(i, j)$  whose  $i$  is the current node, set its value as the smaller one between the arrival time at  $j$  calculated in Step 4 and the value set for  $(i, j)$ . Go to Step 3.

### 3.4. Assignment algorithm

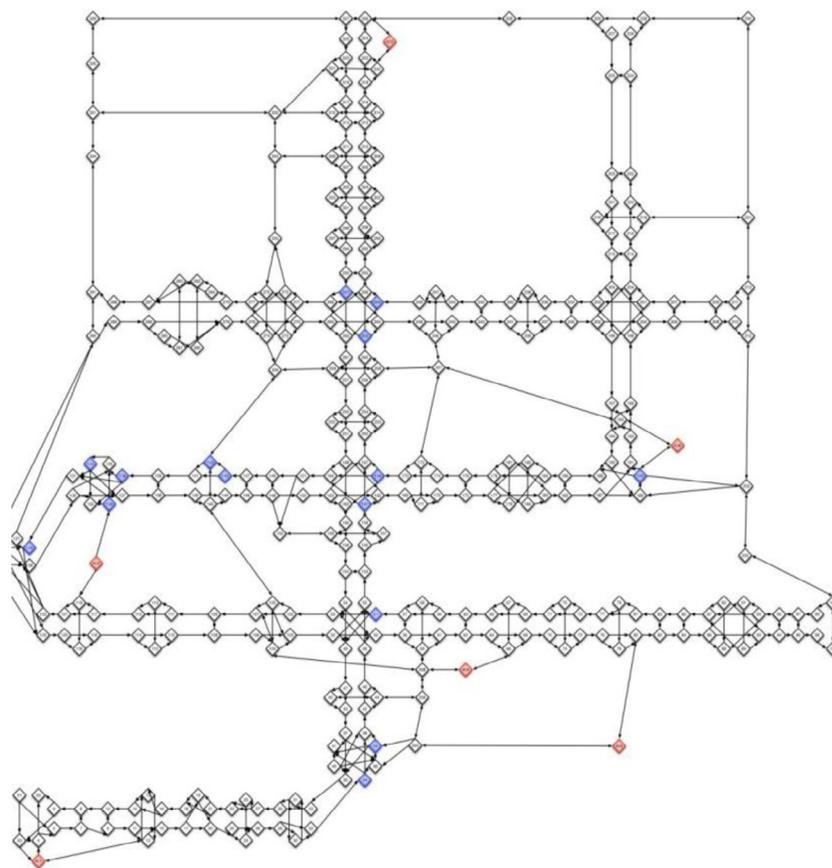
After calculating all the shortest paths between depots and demand points, an assignment algorithm needs to be implemented to determine which vehicles from a depot will travel to a demand point to meet the demand. There are several variants of APs (Pentico, 2007), e.g. generalized AP, bottleneck AP, quadratic AP, and semi-AP. A semi-AP is a modified classic AP which differs in the disproportionality between demand points and depots. The TDVRP is a semi-AP except that a vehicle is not required to be assigned to a demand point. The Hungarian method (Kuhn, 1955) is used as the assignment algorithm to solve the TDVRP.

## 4. Experiments and validation

### 4.1. Case study of a small road network

A road network in Midwest of the United States of America (Figure 2) is analyzed to validate the TDSP and assignment algorithms. In this case study, a no-notice evacuation prompts the need for transportation agencies to deploy vehicles and personnel to specific locations for traffic control. The network has 346 nodes (diamonds in Figure 2), out of which six are depots (red diamonds) and 15 are demand points (blue diamonds). The depots are locations where a state or federal transportation agency, e.g. the state department of transportation, dispatches service vehicles and equipment. The demand points are locations where the agency identifies a need for traffic control during a no-notice evacuation event. Overall, there are 654 roads (arcs in Figure 2) connecting all the nodes; thus the

**Figure 2. A road network in Midwest of the United States of America.**



**Table 1. Algorithms run times (in seconds) for the road network in Figure 2**

Planning period (min)	GAMS	VBA® node labeling	VBA® arc labeling
120	191	9	13
240	449	11	13
360	681	13	13
480	872	15	15
1,440	1,954	33	24

mean degree is  $\frac{654 \times 2}{346} = 3.78$ . There are a total of 92 vehicles available at six depots of a public transportation agency. Each time interval is one minute. The TDSP and assignment algorithms are implemented in VBA® and the GAMS (GAMS Development Corporation, 2013). All computation results are obtained using a Windows 7 ×64 PC, as described previously. Table 1 summarizes run times.

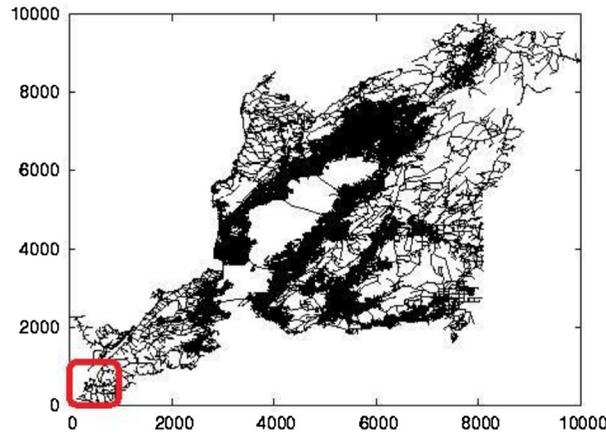
Algorithms implemented in VBA® and the GAMS provide the same optimal solutions and values. Table 1 shows, however, that there is a substantial advantage in using VBA® to identify the optimal solution and value for real-time TDVRPs. The GAMS employs a suite of solvers, e.g. CPLEX, and algorithms, e.g. simplex and branch-and-cut algorithms, to solve linear programming and mixed integer programming problems, but lack efficiency in computer memory management. To compute the results in Table 1, both the TDSP algorithm with arc labeling and assignment algorithm are coded in the GAMS to identify the optimal solution and value. It takes even more computation time when models in Equations (1) and (2) are directly used in the GAMS to find the optimal solution and value. The TDSP and assignment algorithms implemented in VBA® directly manipulate data and are customized to solve the TDVRPs; they are more efficient in terms of run time and computer storage space. Table 1 does not show a substantial difference in run times between node labeling and arc labeling for the TDSP algorithm. The difference between these two methods needs to be further investigated using larger road networks.

#### 4.2. Node labeling and arc labeling in VBA® for a large road network

Figure 3 shows a road network in the City of San Francisco (Brinkhoff, 2002), which includes 174,956 nodes and 223,001 arcs with a mean degree of 2.55 ( $= \frac{223,001 \times 2}{174,956}$ ). A portion of the network (area inside the red box at the lower left corner of Figure 3), including 492 nodes and 559 arcs that connect the nodes, is selected to validate the TDSP and assignment algorithms. The selected portion has a mean degree of 2.27 ( $= \frac{559 \times 2}{492}$ ). Six depots and 15 demand points are randomly and uniformly selected among the 492 nodes. Total 92 vehicles are randomly and uniformly stationed at six depots. Each time interval is one minute. The GAMS requires more than 10 h computing the optimal solution to the TDVRP for this network and sometimes stops unexpectedly due to insufficient computer memory. Run times for VBA® are summarized in Table 2, which does not include run times for the GAMS since they are too large and the authors choose not to complete the runs in the GAMS. The authors choose five different planning periods, each of which specifies a time period during which vehicles must be dispatched to demand points to meet the demand. Run time increases as the planning period increases, indicating more computation time is needed for a larger planning period.

Arc labeling takes more time to compute the optimal solution than node labeling when the planning period is less than 1,440 min. The difference in data processing time between arc labeling and node labeling is relatively insignificant when the planning period is short. Once the TDSP is identified, the optimal solution, i.e. a sequence of nodes on the shortest path, must be determined by backtracking from the destination to origin. Arc labeling is expected to take more time in backtracking since it must find the two end nodes of a visited arc. The advantage of arc labeling becomes evident when the number of nodes or planning period increases. In Table 2, when the planning period is 1,440 min (24 h), node labeling could not compute the optimal solution due to insufficient memory;

**Figure 3. A road network in the City of San Francisco (Brinkhoff, 2002).**



**Table 2. Algorithms run times (in seconds) for the road network in Figure 3**

Planning period (min)	VBA® node labeling	VBA® arc labeling
120	14	26
240	19	28
360	22	30
480	27	32
1,440	N/A	48

run time is not applicable (N/A). It takes 48 s for arc labeling to compute the optimal solution, which is acceptable for real-time TDVRPs.

#### 4.3. Run times of TDSP and assignment algorithms for simulated road networks

The road network in Section 4.1 has a mean degree of 3.78 whereas the two road networks in Section 4.2 have a mean degree of 2.55 and 2.27, respectively. To further validate the algorithms implemented in VBA®, simulated road networks are generated for a given degree distribution. A random network, one of the most studied networks, is described using a degree distribution  $\binom{n-1}{d} p^d (1-p)^{n-1-d}$ , where  $n$  is the total number of nodes,  $d$  is node degree, and  $p$  is the probability that an arc between two nodes exists. The mean degree  $\bar{d} = (n-1)p$ . Random networks were used to describe the structure of the US highway system (Barabasi, 2002; Jeong, 2003). Five random road networks with total 500, 1,000, 1,200, 1,500, and 2,000 nodes are generated using a mean degree of 2.5.

The TDSP and assignment algorithms in VBA® are applied to the five simulated networks and their run times are summarized in Table 3, which reveals several important observations:

- Run times increase as the total number of nodes increases;
- Run times increase in most cases as the planning period increases. This is consistent with results in Tables 1 and 2;
- If node labeling can compute an optimal solution to the TDVRP, it takes less time than arc labeling. This is due to the additional time required for arc labeling to backtrack from the destination to origin and identify the optimal solution;
- Node labeling cannot compute an optimal solution when size of the matrix that stores travel times between nodes becomes too large. For instance, when total number of nodes is 1,500 and 2,000, node labeling cannot compute an optimal solution for any planning period; and

**Table 3. Algorithms run times (in seconds) in VBA® for five simulated road networks**

Number of nodes	Labeling	Planning period (min)				
		120	240	360	480	1,440
500	Node	18	19	24	28	N/A
	Arc	34	36	39	41	62
1,000	Node	50	N/A	N/A	N/A	N/A
	Arc	92	91	96	100	129
1,200	Node	74	N/A	N/A	N/A	N/A
	Arc	117	123	128	133	170
1,500	Node	N/A	N/A	N/A	N/A	N/A
	Arc	234	237	246	252	301
2,000	Node	N/A	N/A	N/A	N/A	N/A
	Arc	388	449	414	410	479

(e) Arc labeling can efficiently compute optimal solutions to the TDVRP for large road networks over a long planning period. The largest run time in Table 3 is 479 s, which is about 8 min and acceptable for real-time TDVRPs.

### 5. Conclusions and future research

This paper develops a new TDSP algorithm with arc labeling and integrates it with the assignment algorithm to identify the optimal solution and value for the TDVRP in real time. The experiments show that the algorithms implemented in VBA® may be used for real-time vehicle routing in large road networks. The case study presented in Section 4.1 implements the algorithms in a no-notice evacuation for a metropolitan area with acceptable performance and does not require above average computing power. Both the TDSP and assignment algorithms are validated using a large road network and five simulated networks. The TDSP with arc labeling may be used for real-time TDVRP over a long planning period. Future research will investigate the performance of the TDSP and assignment algorithms in a variety of real-world road networks with various degree distributions, and explore certain heuristic algorithms, e.g. genetic algorithm, to solve the TDVRP. In addition, other programming languages, e.g. C++, Java, and Julie, will be used to further improve the efficiency of the algorithms.

#### Funding

The authors received no direct funding for this research.

#### Author details

Xin Chen<sup>1</sup>  
 E-mail: [xchen@siue.edu](mailto:xchen@siue.edu)  
 Zhu Zhang<sup>1</sup>  
 E-mail: [zhuzhang1989@gmail.com](mailto:zhuzhang1989@gmail.com)  
 Ryan Fries<sup>2</sup>  
 E-mail: [rfries@siue.edu](mailto:rfries@siue.edu)

<sup>1</sup> Department of Industrial Engineering, Southern Illinois University Edwardsville, Edwardsville, IL 62026-1805, USA.

<sup>2</sup> Department of Civil Engineering, Southern Illinois University Edwardsville, Edwardsville, IL 62026-1800, USA.

#### Citation information

Cite this article as: Development and implementation of algorithms for vehicle routing during a no-notice evacuation, Xin Chen, Zhu Zhang & Ryan Fries, *Cogent Mathematics* (2016), 3: 1163767.

#### References

Almoustafa, S., Hanafi, S., & Mladenović, N. (2013). New exact method for large asymmetric distance-constrained vehicle routing problem. *European Journal of Operational Research*, 226, 386–394. <http://dx.doi.org/10.1016/j.ejor.2012.11.040>

Ando, N., & Taniguchi, E. (2006). Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, 6, 293–311. <http://dx.doi.org/10.1007/s11067-006-9285-8>

Andres Figliozzi, M. (2012). The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review*, 48, 616–636. <http://dx.doi.org/10.1016/j.tre.2011.11.006>

Balas, E., & Toth, P. (1985). Branch and bound methods. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, & D. G. Shmoys (Eds.), *The traveling salesman problem* (pp. 361–401). Chichester: Wiley.

Barabasi, A. L. (2002). *Linked: The new science of networks*. Cambridge, MA: Perseus.

- Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39, 104–118. <http://dx.doi.org/10.1287/trsc.1030.0056>
- Brinkhoff, T. (2002). A framework for generating network-based moving objects. *GeoInformatica*, 6, 153–180. <http://dx.doi.org/10.1023/A:1015231126594>
- Chen, H. K., Hsueh, C. F., & Chang, M. S. (2006). The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42, 383–408. <http://dx.doi.org/10.1016/j.tre.2005.01.003>
- Chen, B. Y., Lam, W. H. K., Sumalee, A., Li, Q., Shao, H., & Fang, Z. (2013). Finding reliable shortest paths in road networks under uncertainty. *Networks and Spatial Economics*, 13, 123–148. <http://dx.doi.org/10.1007/s11067-012-9175-1>
- Clarke, G., & Wright, J. V. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, 568–581. <http://dx.doi.org/10.1287/opre.12.4.568>
- Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). Solution of a large-scale traveling salesman problem. *Operations Research*, 2, 393–410.
- Dreyfus, S. E. (1969). An appraisal of some shortest-path algorithms. *Operations Research*, 17, 395–412. <http://dx.doi.org/10.1287/opre.17.3.395>
- GAMS Development Corporation. (2013). *GAMS Distribution 24.0.2*. Retrieved from <https://www.gams.com/>
- Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32, 2959–2986.
- Haimovich, M., Rinnooy Kan, A. H. G., & Stougie, L. (1988). Analysis of heuristic routing problems. In B. L. Golden & A. Assad (Eds.), *Vehicle routing: Methods and studies* (pp. 47–61). Amsterdam: North Holland.
- Ichoua, S., Gendreau, M., & Potvin, J. Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144, 379–396. [http://dx.doi.org/10.1016/S0377-2217\(02\)00147-9](http://dx.doi.org/10.1016/S0377-2217(02)00147-9)
- Jeong, H. (2003). Complex scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 321, 226–237. [http://dx.doi.org/10.1016/S0378-4371\(02\)01774-0](http://dx.doi.org/10.1016/S0378-4371(02)01774-0)
- Kaufman, D. E., & Smith, R. L. (1993). Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. *Journal of Intelligent Transportation Systems*, 1(1), 1–11.
- Kritzing, S., Doerner, K. F., Hartl, R. F., Kiechle, G. Y., Stadler, H., & Manohar, S. S. (2012). Using traffic information for time-dependent vehicle routing. *Procedia—Social and Behavioral Sciences*, 39, 217–229. <http://dx.doi.org/10.1016/j.sbspro.2012.03.103>
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2, 83–97. [http://dx.doi.org/10.1002/\(ISSN\)1931-9193](http://dx.doi.org/10.1002/(ISSN)1931-9193)
- Laporte, G., Nobert, Y., & Taillefer, S. (1987). A branch-and-bound algorithm for the asymmetrical distance-constrained vehicle routing problem. *Mathematical Modelling*, 9, 857–868. [http://dx.doi.org/10.1016/0270-0255\(87\)90004-2](http://dx.doi.org/10.1016/0270-0255(87)90004-2)
- Laporte, G., Nobert, Y., & Taillefer, S. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, 22, 161–172. <http://dx.doi.org/10.1287/trsc.22.3.161>
- Lecluyse, C., Van Woensel, T., & Peremans, H. (2009). Vehicle routing with stochastic time-dependent travel times. *4OR—Journal of Operations Research*, 7, 363–377.
- Maden, W., Eglese, R., & Black, D. (2010). Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61, 515–522. <http://dx.doi.org/10.1057/jors.2009.116>
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79, 115–129.
- Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176, 774–793. <http://dx.doi.org/10.1016/j.ejor.2005.09.014>
- Rakha, H., El-Shawarby, I., Arafeh, M., & Dion, F. (2006). *Estimating path travel time reliability*. In *Proceedings of 2006 IEEE Intelligent Transportation Systems Conference* (pp. 236–241), Toronto.
- Spliet, R., & Gabor, A. F. (2012). The time window assignment vehicle routing problem. *Erasmus School of Economics (ESE), EI2012\_07*, 1–19.
- Van Woensel, T., Kerbache, L., Peremans, H., & Vandaele, N. (2008). Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research*, 186, 990–1007. <http://dx.doi.org/10.1016/j.ejor.2007.03.012>
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234, 658–673. <http://dx.doi.org/10.1016/j.ejor.2013.09.045>
- Zare-Reisabadi, E., & Mirmohammadi, S. H. (2015). Site dependent vehicle routing problem with soft time window: Modeling and solution approach. *Computers & Industrial Engineering*, 90, 177–185.



© 2016 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

