1-1-2015

Dynamic Vehicle Routing in Emergency Evacuation

Yi Wen

Dynamic vehicle routing in emergency evacuation

By

Yi Wen

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Civil Engineering
in the Department of Civil and Environmental Engineering

Mississippi State, Mississippi

August 2015

Dynamic vehicle routing in emergency evacuation

By

Yi Wen

Approved:

Li Zhang
(Major Professor)

Xiaopeng Li
(Co-Major Professor)

William H. McAnally
(Committee Member)

Mingzhou Jin
(Committee Member)

Qi Zhang
(Committee Member)

James L. Martin
(Graduate Coordinator)

Jason M. Keith
Dean
Bagley College of Engineering

Name: Yi Wen

Date of Degree: August 14, 2015

Institution: Mississippi State University

Major Field: Civil Engineering

Major Professor: Li Zhang

Title of Study: Dynamic vehicle routing in emergency evacuation

Pages in Study: 133

Candidate for Degree of Doctor of Philosophy

Since Hurricane Katrina, extensive studies have been conducted aiming to optimize the transit vehicle routing in the event of an emergency evacuation. However, the vast majority of the studies focus on solving the deterministic vehicle routing problem that all the evacuation data are known in advance. These studies are generally not practical in dealing with real-world problems which involve considerable uncertainty in the evacuation data set. In this dissertation, a SmartEvac system is developed for dynamic vehicle routing optimization in emergency evacuation. The SmartEvac system is capable of processing dynamic evacuation data in real time, such as random pickup requests, travel time change, network interruptions. The objective is to minimize the total travel time for all transit vehicles.

A column generation based online optimization model is integrated into the SmartEvac system. The optimization model is based on the following structure: a master problem model and a sub-problem model. The master problem model is used for routes selection from a restricted routes set while the sub-problem model is developed to progressively add new routes into the restricted routes set. The sub-problem is formulated

as a shortest path problem with capacity constraint and is solved using a cycle elimination algorithm. When the evacuation data are updated, the SmartEvac system will reformulate the optimization model and generate a new routes set based on the existing routes set. The computational results on benchmark problems are compared to other studies in the literature. The SmartEvac system outperforms other approaches on most of the benchmark problems in terms of computation time and solution quality.

CORSIM simulation is used as a test bed for the SmartEvac system. CORSIM Run-Time-Extension is developed for communications between the simulation and the SmartEvac system. A case study of the Hurricane Gustav emergency evacuation is conducted. Different scenarios corresponding to different situations that presented in the Hurricane Gustav emergency evacuation are proposed to evaluate the performance of the SmartEvac system in response to real-time data. The average processing time is 28.9 seconds and the maximum processing time is 171 seconds, which demonstrate the SmartEvac system's capability of real-time vehicle routing optimization.

DEDICATION

I would like to dedicate this research to my parents, my wife, and my daughter.

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to Dr. Li Zhang for his invaluable advice, guidance, and support during my Ph.D. study. Without his persistent help this dissertation would not have been possible.

Many thanks to the other members of my committee, Dr. Xiaopeng Li, Dr. Mingzhou Jin, Dr. William H. McAnally, and Dr. Qi Zhang. Each of you, with your special expertise, provided me with practical advice and continuous encouragement.

I would like to extend my appreciation to my friends and colleagues, Dr. Jizhan Gou, Zhitong Huang, William Fulcher, and Lei Zhang, for your encouragement and companionship.

I would like to thank the faculty, students, and staff in the Civil and Environmental Engineering Department. Financial support was provided by the National Center for Intermodal Transportation for Economic Competitiveness (NCITEC). I would also thank the Coast Transit Authority (CTA) for providing the data and advice.

Finally, I wish to thank my parents, Jiuzhu Wen, Fengyun Zhang, my wife, Xiaojun Wang, and my daughter, Sihan Wen, for your continuous support and encouragement. I appreciate you more than words can express.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background and Motivation

In 2005, transit could have played an important role by assisting in the evacuation of an estimated 150 to 200 vulnerable residents in Gulf Coast region who lacked access to a private vehicle during Hurricane Katrina. A transit evacuation plan was proposed but not carried out effectively due to incident control failure and resources unavailability, e.g. few transit drivers reported to work. In response to the lessons learned from Hurricane Katrina, transit agencies are taking more active actions in evacuating transit-dependent populations from emergency. Federal Transit Administration has established the Advanced Public Transportation Systems (APTS) program to encourage development and implementation of innovative technologies and strategies to improve transit service in emergency. In Los Angeles, transit-dependent people are able to register with APTS over the phone. APTS will call them three times to verify their demand for transportation assistance and to inform them when the evacuation service is on the way. In addition, transit agencies across the nation, including Regional Transportation District (RTD) in Denver, Milwaukee County Department of Public Works Transportation Division (MCTD), Kansa City Area Transportation Authority (KCTA), Maryland's Mass Transit Administration (MTA), and Dallas Area Rapid Transit (DART), are implementing Automatic Vehicle Location (AVL) system to monitor both locations and performances of transit vehicles. Data collected by AVL can be used to monitor schedule adherence and conduct real-time transit vehicle routing.

Other active measures include the following (White, 2008):

- Develop transit emergency evacuation plans coordinated with regional evacuation plans.
- Register transit-dependent people.
- Identify the maximum number of transit-dependent people that could be serviced.
- Consider school buses and drivers for meeting the surge demands of emergency evacuation.
- Develop plans specially for evacuating people with special needs, e.g., the disabled, the elderly.
- Develop standby emergency service contracts to fill the remaining transit service gap.
- Build real time communication among transit drivers, emergency managers, infrastructures, as well as the public.
- Coordinate with state and local department of transportation to provide dedicated lanes to facilitate transit trips in emergency evacuation.

The above actions have illustrated the progress achieved since 2005. However, the potential for transit to play a more significant role in emergency evacuation is still far from being realized. The USDOT (U.S. Department of Transportation) study (USDOT, 2006) evaluated the evacuation plans in the Gulf Coast region. The assessment focused on the role of transit in emergency evacuation in the 33 urbanized areas of Gulf Coast region. In 11 of the 33 urbanized areas, transit was included in the emergency evacuation plans, but only 7 of the plans provided sufficient detail concerning the role of transit in emergency evacuation. There are many factors that limit the role of transit. An obvious limitation is caused by damage to transportation system. For example, two major bridges on U.S. 90 along Mississippi Gulf Coast, the Bay St. Louis Bridge and the Biloxi Bay Bridge, were destroyed completely by Hurricane Katrina, which seriously impeded the emergency evacuation progress. Another limitation comes from severe congestions in peak hours of emergency evacuation. Transit service could be interrupted due to the transportation network capacity shortfall. One of the most serious limitations is unpredictability of evacuation data. Even for a hurricane emergency evacuation with advance notice, when the hurricane will make landfall and what its path is will be remain uncertain in the planning stage. The number of evacuees also largely depends on the size and severity of the hurricane. It could seriously hamper the transit service if the system could not respond to updated evacuation data in real time.

Current fleet management software used by transit agencies, such as RouteMatch and SafePath, are designed for regular operations when the environment and relevant factors are predictable and relatively stationary. These commercial software tools however cannot handle transit operations efficiently in uncertain environment. It is anticipated that in the future the transit agencies need to update both software and hardware in order to efficiently operate the transit in emergency evacuation.

Transit operations in emergency evacuation is defined as a Capacitated Dynamic Vehicle Routing Problem with Pickup and Delivery (CDVRPPD), which is an extension of the traditional Vehicle Routing Problem (VRP). The CDVRPPD involves solving the vehicle routing problem with pickup and delivery in a real-time environment. In the past decade, numerous studies on the VRP have been published. The majority of the VRP literature is dedicated to the deterministic VRP that all the data are known in advance. However, transit operations in a real-world emergency evacuation often involve uncertainties with respect to the locations and demands of unregistered evacuees, road travel time, etc. Thus a practical transit management system should be able to capture real-time evacuation data, and re-optimize transit vehicle routes based on real-time evacuation data. Recent developments in communication technologies make it realizable and affordable to update evacuation data in real time. As real-time evacuation data is available to transit agencies, a fleet management system capable of real-time evacuation data processing and dynamic transit vehicle routing becomes more urgent. In this dissertation, a SmartEvac system is developed for fleet management in emergency evacuation. Features of the SmartEvac system include:

- Real-time evacuation data collection and processing;
- Demand-responsive transit vehicle routing and scheduling;
- Real-time response to transportation network interruptions.

Objectives and Approaches

The objective of this dissertation is to develop a real-time transit management system, called SmartEvac, which can be adopted in emergency evacuations of mid-size cities. It is an intelligent system designed to support more effective delivery of transit service. The SmartEvac system focuses on optimizing the fleet planning, scheduling, and operations.

The SmartEvac system supports advanced demand-responsive transit vehicle routing and scheduling. When a pickup request comes into the system, the SmartEvac system can assist in quickly dispatching appropriate service to the location and rerouting the system to ensure system-wide efficiency.

Transit service is monitored through collection of operational data, including vehicle positions, evacuee data, and traffic conditions. In the events that impact schedule adherence, such as severe congestions and transportation network interruptions, the SmartEvac system could take quick actions, such as rerouting, to improve schedule adherence.

In order to improve transit route running times, a CDVRPPD model is implemented in the SmartEvac system to optimize the total travel time. The CDVRPPD model is based on a master problem – sub-problem structure. The master problem is formulated as a Set Covering (SC) model which is used for routes selection from a restricted routes set. The sub-problem is formulated as an Elementary Shortest Path Problem with Capacity Constraint (ESPPCC) model which progressively adds new routes into the restricted routes set.

A column generation based dynamic algorithm is implemented to solve the CDVRPPD model. The sub-problem is solved using a Cycle Elimination (CE) algorithm. CPLEX is used as the Linear Programming (LP) and Mixed Integer Programming (MIP) solver. In order to evaluate the performance of the proposed algorithms, computational results on benchmark problems are compared to other studies in the literature in terms of solution quality and computation time.

The SmartEvac system is validated through a case study of Hurricane Gustav evacuation in Gulfport, MS. CORSIM simulation is conducted as a proof-of-concept to demonstrate the SmartEvac system's feasibility in a dynamic environment. CORSIM Run-Time-Extension (RTE) is developed as a communication interface which enables data exchange between CORSIM simulation and the SmartEvac system. Different scenarios corresponding to different situations that happened in the Hurricane Gustav emergency evacuation are proposed to evaluate the performance of the SmartEvac system in response to real-time data.

Significance of the Study

The proposed SmartEvac system will enhance the transit service in emergency evacuation. Literature review indicates that transit agencies are playing an important role in evacuating transit-dependent people in emergency evacuation; however, many issues remain unsolved. One of the key issues that affect the evacuation capability is to respond to new pickup requests. The SmartEvac system integrated with state-of-the-art dynamic vehicle routing models and algorithms will effectively handle real-time pickup requests. The dynamic feature of the SmartEvac system gives the dispatcher greatly improved awareness of traffic conditions on the road and the ability to take quick actions to respond to new pickup requests.

The SmartEvac system is expected to have many benefits, including:

- Increased operating efficiency;
- Increased service reliability;
- Increased resilience of transit service in an emergency;

- Improved response to surge demands;
- Improved response to service disruptions;

In summary, the SmartEvac system will improve the efficiency and safety of transit service, which leads to a successful emergency evacuation. Since a poorly executed emergency evacuation risks significant loss of life, particularly among those who are transit-dependent, the SmartEvac system would not only provide efficient and reliable transit service, but also save lives when dynamic factors are involved in the emergency evacuation.

Outline of the Dissertation

This dissertation consists of six chapters. In chapter 1 the background and motivation of the SmartEvac system is described. Chapter 2 presents a review of relevant literature. The development of a CDVRPPD model and its variants are presented in chapter 3. Chapter 4 describes the cycle elimination algorithm to the CDVRPPD model and the SmartEvac system design. Chapter 5 proposes a case study of Hurricane Gustav evacuation. Chapter 6 summarizes the dissertation and subsequent studies.

CHAPTER II

LITERATURE REVIEW

In this chapter, state-of-art evacuation modeling techniques are discussed. A comprehensive literature review of existing vehicle routing problem models, algorithms, and software packages is presented.

Evacuation Modeling

Transit evacuation is playing an increasingly important role following the strikes of severe hurricanes such as Hurricane Katrina and Rita in 2005, and more recently, Hurricane Sandy in 2012. To protect the general public from disaster, it is necessary to develop more advanced evacuation model for evaluating or optimizing transit evacuation operations. Most studies on transit evacuation operations focus on two types of off-line model: simulation model and optimization model. Simulation models are categorized into three groups, microscopic, macroscopic, and mesoscopic, depending on the level of detail at which the traffic information is described. Simulation models allow evacuation managers to develop and compare different evacuation plans for different hypothetical emergency scenarios (Yuan et al., 2006).

Macroscopic Simulation Model

Macroscopic simulation models consider traffic flow as composed of platoons of vehicles, i.e. vehicles with common characteristics are treated as a homogeneous group.

They are mainly developed for evacuation planning purpose. Most macroscopic simulation models are based on dynamic network flow approach (Sheffi et al., 1982; KLD, 1984; Hobeika and Jamei, 1985; Hobeika and Kim, 1998). In the context of emergency evacuation, macroscopic simulation models are often used to analyze the traffic conditions, estimate evacuation times, and generate optimal evacuation routes etc.

NETVAC (Network Emergency Evacuation), developed by Sheffi et al. (1982), is considered as the first evacuation planning simulation model. NETVAC is used for simulating traffic flow patterns and estimating clearance times during emergency evacuations. NETVAC allows the analyst to customize the degree of driver compliance on an intersection specific basis under evacuation conditions. NETVAC also supports dynamic route selection by dynamically adjust the turning movements at each simulation interval according to the traffic conditions. However, the model was specifically designed for nuclear plant accident evacuation, which means the evacuation starts from a single point, and thus all the movements are directed radially outward from the single point rather than a more general direction as a hurricane evacuation.

IDYNEV (Interactive Dynamic Network Evacuation) is a macroscopic evacuation simulation model developed by KLD Associates, Inc. for Federal Emergency Management Agency (FEMA). It is used for estimating evacuation times in nuclear plant accident evacuation. IDYNEV integrates three different models: (a) a deterministic traffic simulation model; (b) an equilibrium traffic assignment model; and (c) a traffic capacity model for intersection approaches (KLD, 1984). The traffic simulation model is capable of rerouting the evacuation traffic if the routes are too congested. Like NETVAC, IDYNEV is designed particularly for nuclear plant accident evacuation and is unable to handle large-scale evacuation such as hurricane evacuation.

Hobeika and Jamei (1985) developed a macroscopic simulation model, MASSVAC (Mass Evacuation), for mass evacuation. MASSVAC uses All-or-Nothing algorithm and Dial's algorithm (Dial, 1971) for dynamic traffic assignment. Traffic volumes are updated during each time interval as new traffic load onto the system. The probability of a particular path being selected is calculated by the product of the probability that each link in the path is used. To improve the evacuation performance, Hobeika and Kim (1998) updated MASSVAC by incorporating user equilibrium (UE) assignment algorithm.

HURREVAC (Hurricane Evacuation) (FEMA, 2013) is a storm tracking and decision support tool developed specifically for hurricane evacuation. HURREVAC combines National Hurricane Center's Forecast Advisories with data from various state HES (Hurricane Evacuation Studies) to estimate the time required to evacuate an area, which assists the local emergency management agency in determining the most appropriate evacuation decision time.

VISUM is a macroscopic simulation software system for traffic analyses. It is used to simulate evacuation plans (Schomborg et al., 2011; ARCADIS, 2011; ARCADIS, 2012), especially when the maximum evacuation time is required. ARCADIS Inc. (2011, 2012) performed VISUM simulations to forecast evacuation times in different scenarios. The VISUM network includes designated evacuation routes plus backup routes in order to accurately reflect the traffic conditions during an evacuation. The potential impacts of the population growth on evacuation time were also analyzed.

Microscopic Simulation Model

Microscopic simulation models focus on modeling of individual vehicle behavior and interaction among vehicles. Microscopic simulation models are generally based on car-following models. They are often used for modeling traffic with complex behavior in an emergency evacuation, such as contra-flow (Lim, 2003), traffic signal preemption (Zhang, 2009), and transit operations (Wen, 2012). Microscopic models are usually resource intensive and thus are only implemented in small networks.

CORSIM (Corridor Simulation) (McTrans, 2014) is a microscopic traffic simulation software package for simulating urban street and freeway traffic systems. It is an integration of two separate microscopic simulation models, NETSIM (Network Simulation) for modeling surface streets, and FRESIM (Freeway Simulation) for modeling freeways. NETSIM, which is the successor of UTCS-I (Urban Traffic Control System) in the 1970s, keeps track of each individual vehicle, including detail characteristics relating to the vehicle within quite complex urban networks. NETSIM provides simulation results in aggregated level. Urbanik and Desrosiers (1981) used NETSIM model to estimate evacuation time for a nuclear plant evacuation. Lim (2003) and Theodoulou et al. (2004) utilized CORSIM to simulate hurricane evacuation with contra-flow strategy. Zou et al. (2005) applied CORSIM simulation for evaluating six evacuation plans for Ocean City hurricane evacuation. Tagliaferri (2005) performed both CORSIM and VISSIM simulations to investigate the effects of the lane reversal plan on hurricane emergency evacuation. ORNL (Oak Ridge National Laboratory) (Bhaduri et al., 2006) developed OREMS (Oak Ridge Evacuation Modeling System), which is an integration of a CORSIM simulation model and a GIS model, to analyze and evaluate large-scale emergency evacuations, conduct evacuation time estimation, and develop evacuation plans. Zhang et al. (2009) proposed a CORSIM model for simulating emergency vehicle operations, including traffic signal preemption, movement on shoulder and red lights, in a hurricane evacuation. NETSIM is also capable of modeling transit operations. Wen et al. (2012) used CORSIM simulation with RTE to simulate transit signal priority and connected vehicle within a large network with over 150 signalized intersections. The impacts of transit signal priority and connected vehicle on transit emergency evacuations were investigated.

Jha et al. (2004) utilized MITSIM (Microscopic Traffic Simulator), which is the core component of MITSIMLab (MITSIM Laboratory), to evaluate emergency evacuation plans for Los Alamos National Laboratory. MITSIM was used as the microscopic traffic simulator to model the emergency evacuation at the operational level. A probabilistic route choice model was implemented to capture drivers' route choice decisions.

VISSIM (PTV, 2009) is a microscopic, time-step, and behavior based simulation model developed for modeling traffic flow, including private vehicles, trucks, transits, railroads, and pedestrians in detail. It has been widely used for evacuation simulation (Chiu et al., 2005; Yuan et al., 2006; Williams et al., 2007; Edara et al., 2010). Han and Yuan (2005) simulated a nuclear power plant evacuation in VISSIM. Dynamic traffic assignment was implemented in the simulation. Yuan et al. (2006) used VISSIM and DYNASMART_P to validate their ODE (One-Destination Evacuation) concept which modified the network by linking each real-world destination point to one common "dummy destination point" with "dummy links". ODE could avoid steps of demand distribution and focus on solving a one-destination dynamic traffic assignment problem. Williams et al. (2007) applied lane reversal operations in the simulation of a hurricane evacuation in VISSIM. The impacts of the lane reversal plan, especially at the contraflow termination point, were evaluated. Edara et al. (2010) built a large-scale hurricane evacuation network in VISSIM. Simulations were performed to evaluate the evacuation routes and locate the major bottlenecks in the network.

TRANSIMS (Transportation Analysis and Simulation System) is an integrated simulation tool specially designed for intermodal transportation analysis, including transit service (Nagel et al., 1997). It has been used for large-scale multimodal evacuation modeling in recent years (Wolshon et al., 2009; Naghawi, 2010; Wolshon and Vinayak, 2012). Naghawi (2010) used TRANSIMS to simulate transit-based evacuation strategies. A TRANSIMS application for New Orleans transit evacuation simulation was developed. Eight evacuation scenarios with varying conditions, such as alternative transit routes and network loading rate, were generated for evaluation. Network average travel time and total evacuation time were used to measure the effectiveness of proposed transit strategies. Wolshon et al. (2009) developed a TRANSIMS application for hurricane evacuation modeling in New Orleans. The application is capable of modeling multimodal mass evacuations at microscopic level. A TransCAD network of the New Orleans region was imported into TRANSIMS. The simulation results were used to evaluate various operational strategies and identify the network bottlenecks.

Mesoscopic Simulation Model

Mesoscopic simulation models compromise between microscopic simulation models and macroscopic simulation models. They simulate individual vehicles with high level of detail, but describe their activities and interactions based on aggregate relationships. The aggregation mitigates calculative burden and lessens computation time. Typical applications of mesoscopic simulation models in the context of emergency evacuation are reviewed as follows.

Dynasmart-P (Dynamic Network Assignment-Simulation Model for Advanced Roadway Telematics - Planning version), which is the planning version of Dynasmart (Mahmassani et al., 1994), utilizes mesoscopic models to represent traffic interactions. Dynasmart-P supports transportation network planning and operation analyses through the use of simulation-based dynamic traffic assignment. It is capable of handling large-scale urban traffic network with up to 89999 nodes (Mahmassani et al., 2004). In recent years, it has been promoted to incident management strategies evaluation (Kwon, 2004; Yuan et al., 2006; Naser and Shawn, 2010). Kwon (2004) used Dynasmart-P simulations to evaluate emergency evacuation strategies on a large-scale network. Dynasmart-P simulations were developed for a hypothetical emergency evacuation in downtown Minneapolis, Minnesota. The model was calibrated using loop detection data. Alternative emergency evacuation strategies in terms of different network configurations were proposed and evaluated. However, the assumption that all the drivers are aware of the network configuration changes and can adjust their routes accordingly is not realistic under real emergency situations. Naser and Shawn (2010) developed a Dynasmart-P application integrated with Cube-Voyager software (Citilabs, 2013), which provided OD matrix for Dyansmart-P, to model flood evacuation at regional level. Different hypothetical emergency scenarios with varying flood locations, levels, and warning times, were modeled using the Fargo-Moorhead metropolitan area data. Traffic controls were modified to facilitate the evacuation operations. The outputs of the Dynasmart-P simulation were used to estimate the evacuation time, measure the effectiveness of the modified traffic control, and evaluate the system parameters, such as driver compliance and trip loading rate.

DynusT (Chiu et al., 2010) is another version of Dynasmart developed for real-time analysis. It have been implemented in various evacuation studies (Chiu et al., 2008; Zhang et al., 2009; Zheng et al., 2010; Songchitruksa et al., 2012). Chiu et al. (2008) deployed and assessed the contra-flow operation in the Central Texas Evacuation network (CTE) in DynusT. The simulation results indicated that the contra-flow operation led to about 14% travel time savings for all evacuees. Songchitruksa et al. (2012) created DynusT simulations for assessing the performance of alternative evacuation strategies, including partial contra-flow and "evaculane", on which evacuation traffic could use the outside paved shoulder as an additional traveling lane during an emergency evacuation, in the context of a hurricane evacuation in Houston, TX. The evaluation results indicated that the "evaculanes" on I-10 and US-290 could provide sufficient capacity to handle high evacuation demand on both routes without the contraflow operation. In addition, the contra-flow plan for I-45 was proved to be adequate to handle high evacuation demand in lieu of fully implemented contra-flow operation. Wang et al. (2014) incorporated contra-flow with VMS (Variable Message Signs) in a hypothetical emergency evacuation. DynasT simulations were developed to evaluate the performance of the strategies. The simulation results demonstrated the combination of contra-flow and VMS improved the evacuation performance more effectively than using only one or none of the two strategies.

Optimization-Based Evacuation Model

Simulation-based evacuation models only answer “what if” questions about the evacuation system change in a virtual environment. The decision-making procedure is limited by the candidate evacuation plans. In practice, the candidate evacuation plans are usually proposed based on experience, which could deviate far from the optimal evacuation plan especially in the case of an emergency evacuation. Consequently, the simulation-based evacuation models require tremendous amount of time for calibration. As an alternative paradigm, optimization-based evacuation models are capable of identifying optimal evacuation strategies in a systematic, self-driven manner. Optimization-based evacuation models are typically written in a mathematical programming form with an objective of minimizing the total evacuation time or maximizing the network traffic throughput. A set of constraints which describe the objects relationships and the system limitations are formulated to define the solution space. However, optimization-based evacuation models usually need much longer computation time than simulation model and thus are only applied to simplified representation of evacuation problems.

Liu et al. (2006) proposed a two-level integrated optimization model to generate a set of evacuation plans for large-scale network evacuation planning. A revised cell transmission formulation, which allows cells in different sizes being connected arbitrarily, was proposed to model the traffic flow conservation and propagation. The objective of the high level optimization is to maximize the total number of vehicles entering the destinations within the specified evacuation duration. The low level of optimization aims to minimize the total travel time including vehicles’ waiting time. The system parameters were optimized separately in those two levels. For example, the demand distribution is optimized in the high level optimization while the turning percentages at certain critical intersections are optimized in the low level optimization. Finally, the optimized system parameters were implemented in CORSIM simulations for evaluation.

Cova and Johnson (2003) proposed a lane-based routing plan to reduce the traffic delay at intersections during an evacuation. The lane-based routing plan was modeled as the minimum cost flow problem with the objective of minimizing total travel distance. The other objective of the model is to reduce conflicts at intersections. The traffic flow was regulated so as to eliminate crossing conflicts at some critical intersections and minimize lane changing along multi-lane arterials. In practice, such constraints can be readily implemented with emergency personnel directions and installations of portable traffic barriers and road signs at intersections. These two objectives of reducing travel distance and reducing conflicts at intersections are usually conflict and hence a parameter for trade-off between these two objectives was set up based on the traffic volumes. The authors used a node-per-lane mathematical graph to represent lane connectivity. This network representation is a finer level of geographic detail than typical node-per-intersection network representation because the latter may conceal important traffic flow details that might cause delays (Ziliaskopoulous and Mahmassani, 1996). The lane-based routing model was formulated as a MIP solved by CPLEX. Paramics simulations were developed to evaluate the proposed lane-based routing plans under different scenarios.

Kim et al. (2008) developed a macroscopic network flow model to optimize contra-flow during evacuations. The network is represented using a mathematical graph with fixed capacity and no partial contra-flow. The authors demonstrated the NP-completeness of the contra-flow problem. Since the computational burden of find the optimal solution increases exponentially with the growth of the problem size, a parameter named "Overload Degree", which addressed two critical factors, the traffic volume and bottleneck capacity, affecting the computation time, was proposed to identify the problem size. Considering the trade-off between the solution optimality and computational efficiency, three solution algorithms were proposed to solve the contra-flow problem with different problem size. An integer programming approach was suggested to solve the contra-flow problem with low overload degree. A greedy heuristics, which applies contra-flow based on the link congestion level, was used to solve the contra-flow problem with medium overload degree. For the contra-flow problem with high overload degree, a bottleneck relief algorithm was developed to solve the problem by iteratively applying contra-flow to the system bottleneck.

Peeta et al. (2011) discussed the dynamic routing problem in the context of emergency evacuation. The study focuses on identifying the paths for evacuating the distressed population from the affected area and delivering relief supplies to the affected area. The first task is accomplished using a K-shortest paths routing module. The K-shortest paths routing module can generate K shortest paths using K-label-setting algorithm. It provides flexible options for the evacuation managers to distribute tremendous evacuees to the K shortest paths so that if a route is not accessible due to interruptions, other candidate routes are still available. The resource delivery task is solved by a multi-stop routing module. This resource delivery problem is considered as multiple shortest path problems when the stops have fixed sequence. However, if there is no fixed order for stops, then the resource delivery problem is converted to classical traveling salesman problem, which is NP-hard. Both modules are implemented on an ArcGIS platform which has an ArcToolbox for solving these two problems. The system is also integrated with TrafficWise, which is a web-based traffic information system hosted by INDOT (Indiana Department of Transportation), for real time traffic information updates.

Stepanov and Smith (2009) designed a system for traffic assignment with stochastic arrivals in the context of an emergency evacuation. The system firstly generated K outgoing paths for each source in the O-D matrix using K-shortest path algorithm. This procedure is executed in the ArcGIS module. To prevent blocking, a maximum arrival rate on a link is calculated on the condition that the blocking probability on the link will not exceed a threshold value. Each link in the network is defined as an *M/G/c/c* state dependent queuing system. Given the maximum arrival rate, the expected travel time on a link can be calculated using *M/G/c/c* queuing delay model. The optimization module of the system is a multi-objective integer programing model which aims to minimize the total travel distance as well as the total clearance time. The evacuation demand at each source is distributed optimally to the K shortest paths considering the two objectives simultaneously. After generating the evacuation plan, a simulation model named MGCCSimul was applied to evaluate it. The MGCCSimul also considers a link as an *M/G/c/c* state dependent queue with Poisson arrivals, general service rate, and limited capacity. The model dynamically updates the service rate on each link and outputs performance measures, i.e. total clearance time and total travel time.

Vehicle Routing Problem

The vehicle routing problem was first introduced by Dantzig and Ramser (1959), as a generalization of the well-known traveling salesman problem. The VRP consists of finding a set of optimal routes for a fleet of vehicles to service a set of customers, subjected to certain constraints. The classical VRP with its variants, such as the Capacitated Vehicle Routing Problem (CVRP), the VRP with time windows (VRPTW), and the VRP with Pickup and Delivery (VRPPD), have been extensively studied for over 50 years. Current exact algorithms are able to solve the CVRP with a size limit of 50 – 100 customers depending on the customers' distribution and the response time requirement. However, in terms of dynamic vehicle routing problem, most of studies focus on heuristic algorithms and no existing exact algorithms have been successfully applied to the vehicle routing problem in the context of an emergency evacuation.

Exact Algorithms

Exact algorithms to solve the VRP include the branch-and-bound, the cutting plane, column generation, and the branch-and-price algorithms. A brief review of the exact algorithms is provided in this section.

The column generation algorithm is an efficient algorithm for solving large scale linear programs. It has been widely applied to the VRP and its variants by many researchers. Agarwal et al. (1989), Hadjiconstantinou et al. (1995), and Bixby (1998) developed column generation algorithms for general VRP. Desrochers et al. (1992) applied column generation algorithm on the VRPTW. Jin et al. (2008) proposed a column generation approach to solve the VRP with split delivery (VRPSD). The basic idea of column generation is to iteratively generate a subset of columns and push them into the basis such that the inclusion potentially improves the objective function. The column generation algorithm can be combined with the branch-and-bound algorithm, which is called branch-and-price algorithm. The branching occurs when no columns can enter the basis and the LP solution is not integer.

Desrochers et al. (1992) presented a dynamic programming based optimization algorithm for the VRPTW. The VRPTW is formulated by a set covering form in which the path has not to be elementary. The LP relaxation of the SC model is solved by column generation. The pricing sub-problem, which is the Shortest Path Problem with Resource Constraints (SPPRC), is solved by a label correcting algorithm in which labels are created through a "pulling" process. Two sets of labels were generated for the states at each node. The first set of labels provides an upper bound while the second set of labels relates to a lower bound on the cost of a path associated with a state at each node. The algorithm computes the cost associated with a state at a node by progressive refinement of lower and upper bounds on its value. In addition, a 2-cycle elimination procedure was accomplished by a duplication of the labels. This procedure could tight the relaxed state space by eliminating all cycles of length two. The LP solution is then used in a branch-and-bound algorithm to solve the integer SC model. The algorithm has a pseudo polynomial complexity.

Feillet et al. (2004) proposed an exact algorithm for the Elementary Shortest Path Problem with Resource Constraint (ESPPRC). The algorithm is adapted from Desrochers' (1988) label correcting algorithm. A new resource, which indicates if a label of a node is extendable to another node, is created to enforce the elementary path constraint, as proposed by Beasley and Christofides (1989). The label correcting method is improved by introducing the new resource in the dominance rule. This method could decrease the number of states to be explored and hence reduce the computational complexity. The drawback of the method is that the complexity is strongly related to the graph structure, the number of the nodes, and the tightness of resource constraints.

Righini and Salani (2008) developed a label setting algorithm for the ESPPRC. The traditional label setting algorithm is improved by two new methods. The first method is a bi-directional search with resources bounding in which states are extended both forward from a start node to its successors and backward from a destination node to its predecessors. Then all the forward states and backward states at a node are joined, subject to resource constraints, to make feasible routes. Therefore, states are not extended if at most half of the available amount of resources has been used. This method could effectively reduce the number of states in the solution space. The second method is a combination of bi-directional search with state space relaxation. In this algorithm, the state space is relaxed to allow cycles with length more than two. The path found from the relaxed state space is guaranteed to be feasible regarding to the resource constraints but it is not guaranteed to be elementary. They also provided branch-and-bound strategies to eliminate cycles in order to solve the ESPPRC to optimality.

The pricing sub-problem in the column generation scheme was also stated as Traveling Salesman Problem with Profits (TSPP) by Feillet et al. (2005) in a comprehensive survey. TSPP is considered as a bi-criteria TSP with two opposite objectives, one is to maximize the benefits collection at each vertex, which push the salesman to travel, and the other one is to minimize the travel cost, which prevent the salesman from traveling. The two objects constitute the price of visiting a vertex.

Generally, TSPP is divided into three categories based on the way the two objectives are presented. (1) Profitable Tour Problem (PTP) by (Dell'Amico et al., 1995) in which both objective are combined in the objective function. (2) Orienteering Problem (OP) by (Golden et al., 1987) in which the travel cost is formulated as a constraint. (3) Prize-Collecting TSP (PCTSP) by (Balas, 1989) in which the profits is stated as a constraint. Solution approaches for TSPP were summarized into three groups: (1) exact algorithms; (2) classical heuristics; and (3) meta-heuristic procedures. The performance and applicability of the approaches were identified for different TSPP applications.

Pradhan and Mahinthakumar (2012) designed parallel computing technique for solving shortest path problem. Two graph search algorithms, Dijkstra algorithm and Floyd-Warshall algorithm, are implemented in the parallel computing framework. The Floyd-Warshall algorithm is decomposed in two ways for parallel computing. (a) The task of finding all-pair shortest path is decomposed into multiple single source shortest path problems. These smaller tasks are assigned to each processor. (b) The input distance matrix is decomposed by using a striped row-wise decomposition in addition to (a). Only a portion of the distance matrix (rows) is allocated to each processor. Each processor solves a single source shortest path problem for the sources corresponding to the assigned rows. Communications is accomplished by using the MPI (Message Passing Interface) library.

Meta-heuristics

Alba and Dorronsoro (2004) applied Cellular Genetic Algorithm (cGA), which is a subclass of traditional Genetic Algorithm (GA), to solve basic VRP with the objective of minimizing travel time. The most significant difference between cGA and GA is that the former constructs the population by using the concept of neighborhood, so that individuals can only interact with their neighbors in the population. Chromosomes, which constitute the population, are structured in a two dimensional toroidal grid. Each chromosome is included in a sub-population, like a cellular, which contains the chromosome itself plus four neighborhood chromosomes from North, East, West, and South direction. Hence groups are overlapped in the toroidal and adjacent cellular share neighborhoods. cGA's evolution, such as crossover, mutation, are operated within each cellular. The overlap of the neighborhood provides an implicit mechanism of migration. The best chromosomes spread more smoothly through the whole population that GA. cGA controls the dispersion of the best chromosomes by modifying the size of overlap. In addition to general GA evolution operations, local search techniques, including 2-Opt and λ-interchange, are performed to refine solutions. In comparison with Christofides and Mingozzi's benchmarks, cGA is always capable of locating the optimum of the tested problems within shorter computation time.

Schwardt and Dethloff (2005) developed a variant of Kohonen's algorithm to solve a deterministic, single-depot, capacitated multi-vehicle routing problem. Kohonen's algorithm was based on a neural network including two layers. Weights, which were Euclidean distances between nodes and customer demands, were assigned to the links between the layers. The neural network used self-organization approaches to construct the vehicles mapping and simultaneously generate feasible solutions to the location-routing problem.

Goel and Gruhn (2005) worked on real-life vehicle routing problem with randomly generated demands after planning starts. They considered a diversity of practical constraints, such as time window restrictions, a heterogeneous vehicle fleet, vehicle compatibility constraints, etc. To cope with the complexities of the problem, they improved the Large Neighborhood Search method by using fast insertion methods as the search algorithm. Two insertion methods were developed. One is a sequential insertion method in which unscheduled transportation requests were randomly chosen and all feasible insertion possibilities were considered. The second one was auction method in which vehicles only considered the unscheduled transportation request with low incremental cost. The second method was used for the vehicle routing with time window.

Dynamic Vehicle Routing Problem

Creput et al. (2011) discussed a DVRPTW (Dynamic Vehicle Routing Problem with Time Window) application in a telemedicine system. The application is designed for medical emergency management. The application has the ability to handle emergency calls in real time. They developed an optimizer called Dynamic Optimization System (DOS) to solve the DVRPTW. The optimizer has a 2-level architecture. The top level implements a global meta-heuristic strategy to manage the lower level heuristic solvers. The lower level consists of several existing heuristics, such as 2-opt, local search, and neural networks, to solve a typical VRP. The main algorithm in the lower level is based on local search and self-organizing maps (SOM), which are embedded into an evolutionary algorithm framework. The main algorithm could handle new customer requests by neighborhood search, which costs significantly less computation time than traditional exact algorithms.

Chen et al. (2006) proposed a dynamic model to for the DVRPTW. The dynamic model consists of a series of static vehicle routing problems over the planning horizon.

Column generation is implemented for solving the static problem at each decision epoch. Insertion based heuristic is used to generate new columns based on existing columns. However, the algorithm cannot prove that the optimal solution to the RMP (Restricted Master Problem) is also an optimal solution to the master problem because the pricing sub-problem is not guaranteed to be solved to completion using an insertion method.

The random arrival of customer requests during the operation is considered as the most common dynamism in DVRP. Lund et al. (1996) measured the degree of dynamism using the ratio between the number of the random requests and the total number of requests in the operation. Larsen (2000) evaluated the degree of dynamism by the average of the disclosure time of the requests. Since all the pre-defined requests are known at the beginning of the operation, their disclosure time equals to 0. Obviously, the level of dynamism of a problem increases with the disclosure time of the random requests. Ichoua et al. (2007) defined the level of dynamism in the DVRP by two factors, the frequency of the new requests and the urgency of the new requests.

CHAPTER III

MODEL DEVELOPMENT

In this chapter, transit evacuation operations are analyzed at microscopic level. The development of a Capacitated Dynamic Vehicle Routing Problem with Pickup and Delivery model in the context of an emergence evacuation is discussed.

Problem Statement

Given a transportation network in which emergency evacuation is carried out, the dispatching of transit services resembles the Capacitated Dynamic Vehicle Routing Problem with Pickup and Delivery. The transit emergency evacuation process includes sending transit vehicles from the Coast Transit Authority (CTA) to the hurricane prone area to pick up evacuees, updating transit vehicle routes based on real-time evacuee and traffic information, and delivering evacuees to the designated shelters. The evacuee and traffic information are dynamic in nature. The CTA provides dial-a-ride service that allows evacuees to call in requesting on-site pickup over the whole evacuation process. The dispatcher doesn't have any knowledge of future pickup requests. The information of a real-time pickup request, including its location and demand, become known from the moment it comes into the system. Then the problem is to assign the most appropriate vehicle to the new request. Routes are formed before evacuation but updated dynamically in response to real-time information updates, including new pickup request, travel time change. Comparing to the static vehicle routing problem in which demands are known before the evacuation, the dynamic feature gives more freedom to the evacuees while, at the same time, bringing more challenge to transit agencies.

Developing a Model

The problem described above can be formulated as a special case of Capacitated Dynamic Vehicle Routing Problem with Pickup and Delivery. It consists of two types of problems: a) A static CVRPPD (Capacitated Vehicle Routing Problem with Pickup and Delivery) in the planning stage of an emergency evacuation; and b) A dynamic CVRPPD after the emergency evacuation starts.

Basic CVRPPD Model

In the planning stage of an emergency evacuation, pre-registered evacuees' information, including their demands and locations, are known in advance. Despite of the dynamic factors, such as travel time fluctuations, the CVRPPD is assumed to be static in this stage. The classical CVRPPD generalizes the traveling salesman problem. Thus it is NP-hard. The CVRPPD is defined on a directed graph $G = (V, A)$, where V is the set of vertices and A is the set of arcs. S denotes the set of depots. N denotes the set of pickup points and M denotes the set of shelters, both of which are considered as customers with pickup and delivery demands. Therefore the graph consists of $|N|+|M|+|S|$ vertices that $V = N \bigcup M \bigcup S$.

The set of arcs, A, represents direct connections among the vertices. A non-negative cost c_{ij} is assigned to each arc $(i, j) \in A$. Arc cost c_{ij} generally represents the travel time going from vertex i to vertex j, which corresponds to the shortest path from vertex i to vertex j, and consequently the cost matrix satisfies the triangle inequality. Satisfying the triangle inequality, $c_{iz} + c_{zj} \geq c_{ij}$ for all $i, j, z \in V$, implies that any removal of pickup requests from a feasible route will reduce the route cost and any insertion of pickup requests to a feasible route will increase the routes cost. Using self-loop is not allowed by imposing $c_{ii} = +\infty$, for all $i \in V$. The graph is directed with asymmetric cost matrix. This is realistic especially in the case of an emergency evacuation where outbound traffic is usually much heavier than inbound traffic.

There are certain restrictions imposed on the graph as shown in Figure 3.1. The restrictions are written in the form of $i \not\to j$, where i and j denote the nodes which constitute a restricted link. For example, $S \not\to M$ restriction indicates that a vehicle cannot travel from a depot to a shelter, which means that a route has to pass at least one pickup point before ending at a shelter. Other restrictions including $M \not\to M$, which denotes a vehicle cannot travel among shelters, $M \not\to N$, which denotes a vehicle cannot travel from a shelter to a pickup point, and $N \not\to S$, which denotes a vehicle cannot travel from a pickup point to a depot, are added according to the real emergency evacuation situations. These restrictions theoretically turn the network into an incomplete graph where some of the arcs are restricted. These restrictions can be accomplished by assigning 0 to the decision variables corresponding to the usage of the restricted arcs in the model. An alternative way to represent these restrictions is to impose a very large positive value to the travel cost on restricted arcs. In this dissertation, the latter is used because it is easier to implement.

Each pickup point i has a deterministic non-negative demand d_i. It is assumed that d_i is less than or equal to the vehicle capacity. If d_i is larger than the vehicle capacity, the pickup point i will be divided into multiple pickup points, which coincide and have less demand than the vehicle capacity. For depots and shelters, their demands are fictitiously set to 0. There is also a fixed service time c_s associated with each pickup point i. The service time represents the time needed for loading and unloading. The service time is included in the travel cost c_{ij} associated with each arc. Based on CTA's experience, the service time at a pickup point is normally 2.5 minutes on average. The capacity of each shelter is assumed to be unlimited, which conforms to the actual situation. Therefore, any one of the shelters can accommodate all the evacuees in the network.

A homogenous fleet of transit vehicles K with identical capacity Q services the pickup points and shelters. The fleet size is infinite. Q must be larger than or equal to the sum of all the demands on the route assigned to vehicle k. Overload is not permitted. Each pickup point is serviced exactly once. The service includes scheduled pickup for registered evacuees, dial-a-ride to unregistered evacuees, and delivery to a designated shelter. A precedence constraint which regulates that all the pickup points must be served before any shelter is imposed on the route. The CVRPPD involves the design of a set of minimum cost routes that originate at a depot in S and terminate at a shelter in M after picking up all the evacuees. Practically, vehicles don't have to be back to the depot after arrival at a shelter. However, in order to form a complete route, a set of dummy arcs linking from the shelters to the depots with zero travel cost are introduced to replace the original arcs. Then, each vehicle can go back to the depot after delivery at a shelter via the dummy link. In this case, a directed cycle is associated with a vehicle route.

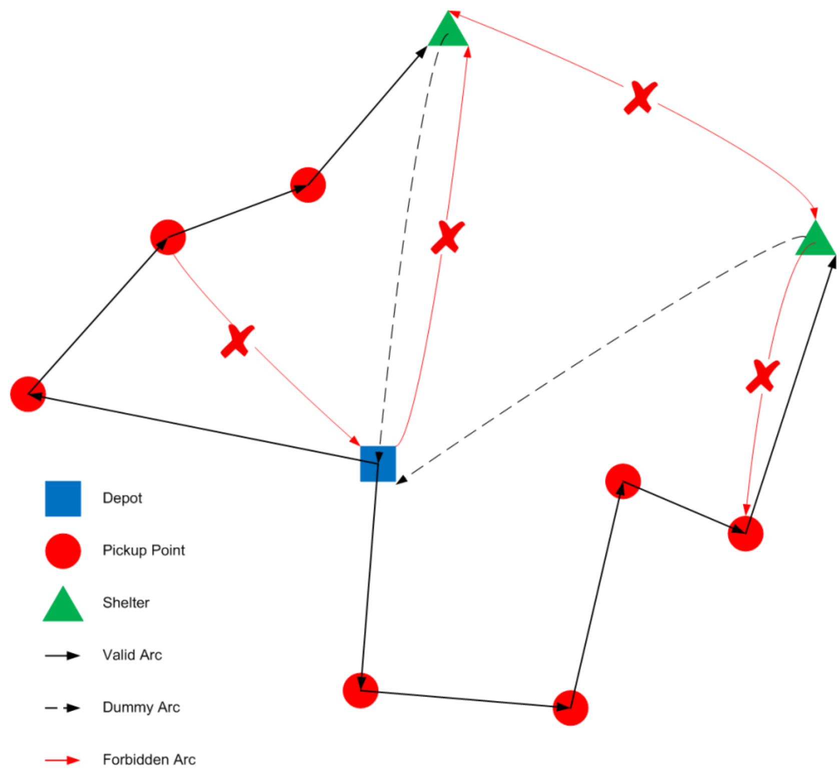

Figure 3.1 A Simple Representation of Vehicle Routes

Figure 3.1 shows a sample of vehicle routes in the network. The blue square denotes a depot. A vehicle starts from the depot and then picks up evacuees at the pickup points which are represented by the red dots. After pickup, the vehicle will deliver the evacuees to a shelter which is represented by the green triangle. The black solid line with arrow denotes the arc which forms a vehicle route. The black dotted line with arrow denotes the dummy link which connects a shelter to a depot. There is no cost associated with a dummy link. The red solid line with arrow denotes the arc which is restricted in the model.

For ease of reference, notations are summarized as follows. Particularly, the set of depots, *S*, contains only one element since the scope of this study is to solve the CVRPPD with single depot.

G = A graph represents the transportation network.

V = Set of vertices in *G*.

A = Set of arcs in *G*.

M = Set of shelters, $M = \{1, 2, \ldots, m\}$.

N = Set of pickup points at the beginning of evacuation,

$N = \{m+1, m+2, \ldots, m+n\}$.

S = Set of depots of all vehicles, $S = \{0\}$.

K = Set of a fleet of vehicles, $K = \{0, 1, 2, \ldots, k, \ldots\}$.

c_{ij} = Travel cost, $\forall (i, j) \in A$.

d_i = Demand at pickup point *i*, $\forall i \in N$.

u_i^k = Vehicle *k*'s load after visiting pickup point *i*, $\forall i \in N, \forall k \in K$.

Q = Vehicle capacity.

The CVRPPD is mathematically formulated as an integer linear programming model by (3.1) – (3.12). The set of decision variables is defined as x_{ij}^k. For each arc $(i, j) \in A$, the integer variable x_{ij}^k indicates whether *(i, j)* is traversed by vehicle *k* in the solution.

$$x_{ij}^k = \begin{cases} 1, \text{ if vehicle } k \text{ travels directly from vertex } i \text{ to vertex } j, \\ \quad \forall i \in V,\ \forall j \in V,\ i \neq j,\ \forall k \in K \\ 0, \text{ otherwise} \end{cases}$$

$$\min z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V \setminus i} c_{ij} x_{ij}^k \tag{3.1}$$

$$\sum_{k \in K} \sum_{j \in V \setminus S} x_{ij}^k = 1, \ \forall i \in N \tag{3.2}$$

$$\sum_{i \in V} x_{iz}^k - \sum_{j \in V} x_{zj}^k = 0, \ \forall z \in V, \ z \neq i, \ z \neq j, \ \forall k \in K \tag{3.3}$$

$$\sum_{j \in N} x_{ij}^k = 1, \ i \in S, \ \forall k \in K \tag{3.4}$$

$$\sum_{i \in N} \sum_{j \in M} x_{ij}^k = 1, \ \forall k \in K \tag{3.5}$$

$$\sum_{i \in M} x_{ij}^k = 1, \ j \in S, \forall k \in K \tag{3.6}$$

$$x_{ij}^k = 0, \ i \in S, \forall j \in M, \forall k \in K \tag{3.7}$$

$$x_{ij}^k = 0, \ j \in S, \forall i \in N, \forall k \in K \tag{3.8}$$

$$x_{ij}^k = 0, \ \forall i \in M, \forall j \in N \bigcup M, \forall k \in K \tag{3.9}$$

$$u_i^k - u_j^k + Q x_{ij}^k \leq Q - d_j, \ \forall i, j \in N, \ i \neq j, \ \forall k \in K, \ such \ that \ d_i + d_j \leq Q \tag{3.10}$$

$$d_i \leq u_i^k \leq Q, \ \forall i \in N, \ \forall k \in K \tag{3.11}$$

$$x_{ij}^k \in \{0,1\}, \ \forall i, j \in V, \forall k \in K \tag{3.12}$$

The objective function (3.1) is to minimize the total travel cost. The in-degree constraints (3.2) ensure that each pickup point is visited once and only once. Route continuity is enforced by the constraints (3.3) as once a vehicle arrives at a pickup point, it has to leave the pickup point. The constraints (3.4), (3.5), and (3.6) indicate that each vehicle leaves an depot exactly once; after picking up all the evacuees on its route it has to visit a shelter once and only once; and finally travels back to the depot, respectively.

The constraints (3.7), (3.8), and (3.9) are connectivity constraints indicating that the arcs from the a depot to a shelter, arcs from a shelter to a pickup point, arcs among shelters, and arcs from a pickup point to a depot, are restricted, respectively. The constraints (3.10) and (3.11) are called polynomial cardinality constraints (Christofides et al., 1979) that impose both sub-tour elimination and the vehicle capacity requirements. Constraints (3.12) are the integrality constraints.

Dantzig-Wolfe Decomposition

For integer linear programming, there are three main indicators of the difficulty, the number of constraints, the number of variables, and the integrality gap. For the CVRPPD model represented by (3.1) – (3.12), obviously it has an exponential number of constraints which make it difficult to solve. Hence, it is desirable to use the knowledge about the problem structure to reformulates the integer program into another equivalent problem which is more manageable for the Simplex Method. Dantzig–Wolfe decomposition is originally developed by George Dantzig and Philip Wolfe (1960) for solving large integer program with special structure. The decomposition, which is closely connected to column generation, is applicable to an integer program with a block angular form as shown in Figure 3.2.

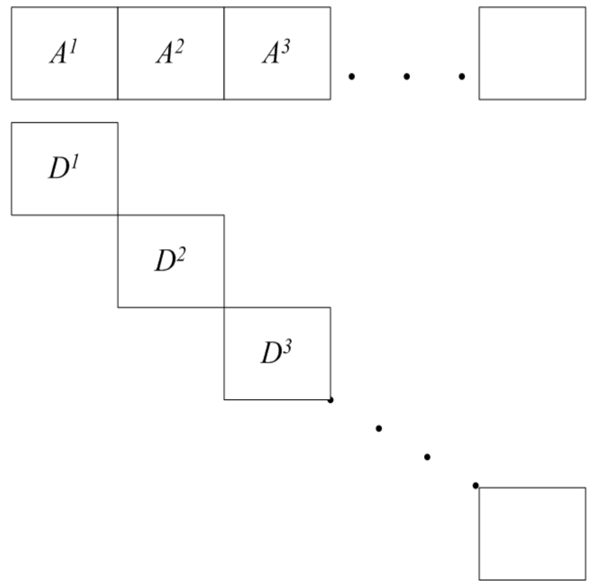

Figure 3.2 The Structure of an Integer Problem with Block Angular Form

The constraints of an integer problem with block angular structure are typically divided into two groups. First, a set of constraints are identified as connecting constraints in which variables are correlated. The sub-matrices of connecting constraints are represented by $A^i,\ i = 1, 2, 3, \ldots$. Second, the remaining constraints are grouped into independent blocks of constraints such that if a variable has a non-zero coefficient in one block, the variable will not have a non-zero coefficient in another block. The sub-matrices of independent blocks of constraints are represented by $D^i,\ i = 1, 2, 3, \ldots$. Consider an integer program (IP) with a block angular form as follows:

$$\min \sum_{i \in B} c^i x^i \tag{3.13}$$

$$\sum_{i \in B} A^i x^i \le b_1 \tag{3.14}$$

$$D^i x^i \le b_2^i,\ \forall i \in B \tag{3.15}$$

$$x^i \in \mathbb{Z}^i_{0+}, \ \forall i \in B \tag{3.16}$$

Where, B is the set of blocks and $\mathbb{Z}^i_{0+}$ is the $|i|$ dimensional domain of non-negative integers. Constraints (3.14) are the connecting constraints where the blocks depend on each other. Constraints (3.15) are $|B|$ independent blocks of constraints. Constraints (3.16) are the integrality constraints. Constraints (3.15) and (3.16) can be combined into the set X^i, which redefine the domain of x^i.

$$X^i = \left\{x^i \in \mathbb{Z}^i_{0+} : D^i x^i \le b^i_2\right\}, \ \forall i \in B \tag{3.17}$$

Assuming that the set X^i is a finite integer set, every point x^i can be presented as a convex combination of its extreme points (Minkowski-Weyl Theorem) such that,

$$x^i = \sum_w^{W^i} \lambda^i_w x^i_w, \ \forall i \in B \tag{3.18}$$

$$\sum_w^{W^i} \lambda^i_w = 1, \ \forall i \in B \tag{3.19}$$

$$\lambda_w \in \{0,1\} \tag{3.20}$$

Where, W^i is the set of extreme points in the domain of X^i and w is an extreme point in X^i. λ_w is binary variable. Substitute x^i into the original IP. Then the IP master problem (IPM) is obtained.

$$\min \sum_{i \in B} \left(\sum_w^{W^i} c^i \lambda^i_w x^i_w \right) \tag{3.21}$$

$$\sum_{i \in B}\left(\sum_{w}^{W^i} A^i \lambda_w^i x_w^i\right) \leq b_1 \tag{3.22}$$

$$\sum_{w}^{W^i} \lambda_w^i = 1,\ \forall i \in B \tag{3.23}$$

$$\lambda_w^i \in \{0,1\},\ \forall i \in B \tag{3.24}$$

The decomposition of the IP into the IPM decreases the number of constraints but increase the number of variables exponentially. Define that $A^i,\ i \in B$ are matrices of size $u_1 \times v_1^i$ and $D^i,\ i \in B$ are matrices of size $u_2^i \times v_2^i$. Table 3.1 shows the number of variables and constraints variation before and after the decomposition.

Table 3.1 Number of Variables and Constraints Before and After Decomposition

Formulation	Number of Variables	Number of Constraints
IP	$\sum_{i \in B} v_1^i$	$u_1 + u_2^i$
IPM	$\sum_{i \in B} \lvert W^i \rvert$	$u_1 + \lvert B \rvert$

Table 3.1 indicates that the decomposition reduces the number of constraints from $u_1 + u_2^i$ to $u_1 + |B|$ since there is only one constraint for each of block in B after decomposition. However, the number of variables may increase exponentially after decomposition. For example, assuming that block i is an unit cube, the domain of x^i is $X^i = \{x^i \in \mathbb{Z}_{0+}^i : x^i \leq 1\}$. Then, after decomposition the number of variables becomes $2^{|i|}$ comparing to $|i|$. So the decomposition redirect the number-of-constraint difficulty towards the number-of-variable difficulty.

For large scale IP, the decomposed model is too large to consider all the variables explicitly. Since for the integer programs solved by the simplex algorithm, most columns are inactive at each step. In such a scheme, a RMP formulation, which contains the currently active columns, iteratively utilizing sub-problems to generate columns for entry into the active columns set, is applied to the IP.

Column Generation Model

Since it is extremely difficult to consider all the variables explicitly when the problem size is large, column generation approach, which represents a generalized application of Dantzig-Wolfe decomposition (DWD), is proposed to solve large integer problems by working with only a subset of variables. The column generation approach is very flexible that the algorithm can be early terminated when an acceptable lower bound is obtained, which is suitable for real time applications.

Master Problem Model

The column generation is based on a master problem and sub-problem structure. For the CVRPPD problem defined through (3.1) – (3.12), the constraints (3.2) are considered as the linking constraints in a DWD scheme which connect the vehicle routes while the remaining constraints (3.3) – (3.12) are associated with individual vehicle. The constraints (3.3) – (3.12) define the domain of individual vehicle route generation, which is the sub-problem. Let R^k be the set of feasible routes traveled by vehicle k and r represents an elementary route in R^k. Let x_{ijr}^k be a binary variable defined as follows.

$$x_{ijr}^k = \begin{cases} 1, \text{ if vehicle } k \text{ travels directly from } i \text{ to } j \text{ on path } r, \\ \quad \forall i \in V,\ \forall j \in V,\ i \neq j,\ \forall r \in R^k,\ \forall k \in K \\ 0, \text{ otherwise} \end{cases}$$

Each variable x_{ij}^k in the IP can be represented by a combination of x_{ijr}^k. The decision variable x_{ij}^k is rewritten by (3.25) – (3.27).

$$x_{ij}^k = \sum_{r \in R^k} x_{ijr}^k y_r^k,\ \forall k \in K,\ \forall i \in V,\ \forall j \in V \tag{3.25}$$

$$\sum_{r \in R^k} y_r^k = 1,\ \forall k \in K \tag{3.26}$$

$$y_r^k \in \{0,1\},\ \forall r \in R^k,\ \forall k \in K \tag{3.27}$$

Where, y_r^k is binary variable that represents whether vehicle k travels on path r. The cost of route r, c_r^k, and the number of times a pickup point i is visited by vehicle k on route r, a_{ir}^k are defined as,

$$c_r^k = \sum_{i,j \in V} c_{ij} x_{ijr}^k,\ \forall r \in R^k,\ \forall k \in K \tag{3.28}$$

$$a_{ik}^r = \sum_{j \in V \setminus i} x_{ijk}^r,\ \forall r \in R^k,\ \forall k \in K,\ \forall i \in V \tag{3.29}$$

Substitute x_{ij}^k and c_{ij} in (3.1) and (3.2) using (3.25) – (3.29). The reformulated CVRPPD master problem is shown by (3.30) – (3.33).

$$\min \sum_{k \in K} \sum_{r \in R^k} c_r^k y_r^k \tag{3.30}$$

$$\sum_{k \in K} \sum_{r \in R^k} a_{ir}^k y_r^k = 1,\ \forall i \in N \tag{3.31}$$

$$\sum_{r \in R^k} y_r^k = 1, \ \forall k \in K \tag{3.32}$$

$$y_r^k \in \{0,1\}, \ \forall r \in R^k, \ \forall k \in K \tag{3.33}$$

Since the fleet of vehicles is homogenous, the travel cost is only associated with the arc, such that $c_r = c_r^k$ for all vehicle *k*. The route sets $R^k = R$ for all vehicle *k*. Therefore, it is possible to eliminate the index *k* by aggregating vehicle *k*'s parameters on route *r*. The revised model is presented as follows.

$$MP : \min \sum_{r \in R} c_r y_r \tag{3.34}$$

$$\sum_{r \in R} a_{ir} y_r = 1, \ \forall i \in N \tag{3.35}$$

$$y_r \in \{0,1\}, \ \forall r \in R \tag{3.36}$$

Now, the arc-based CVRPPD model has been successfully converted to a route-based Set Partitioning (SP) model (Balinski and Quandt, 1964). Equations (3.34) – (3.36) constitute the master problem of the SP model. Notation a_{ir} is binary variable that equals to 1 if vertex *i* is visited by route *r* and equals to 0 otherwise. Decision variable y_r is binary variable that equals to 1 if route *r* is used in the optimum solution and equals to 0 otherwise. Constraint (3.35) defines that each pickup point *i* is covered by one and only one route *r* in the routes set *R*. The master problem is usually relaxed to a Linear Master Problem (LMP) by replacing the integrality constraint (3.36) with $y_r \in [0,1], \ \forall r \in R$.

The columns represented by the decision variables correspond to the feasible routes. Since the number of columns, |*R*|, exponentially increases with the problem size, it is not practical to explicitly enumerate all feasible routes and solve the master problem as an integer programming problem for all but very small sized problem. For example, a network with n customers has theoretically $e(n!)$ elementary routes when n is sufficiently large. The appealing idea to overcome this difficulty is to work with only a small subset of variables first and then generate new variables as needed. The master problem that consider only a subset of variables is so called RMP. The linear relaxation of the RMP (LRMP) is represented by (3.37) – (3.40). The special structure of the SP model results in a tighter linear programming relaxation than that of the arc-based CVRPPD model.

$$LRMP: \min \sum_{r \in R'} c_r y_r \tag{3.37}$$

$$\sum_{r \in R'} a_{ir} y_r = 1, \ \forall i \in N \tag{3.38}$$

$$y_r \in [0,1], \ \forall r \in R' \tag{3.39}$$

$$R' \subseteq R \tag{3.40}$$

Where, R' is a subset of R. The objective of the RMP is to find a set of optimum cost routes within R' to service the pickup points. In the form of a linear relaxation of the RMP, each decision variable y_r represents the number of times the path r is used in the optimum solution. The decision variable y_r is not necessarily integer. Actually it is possible to be any real number in the interval [0, 1].

Instead of the SP model in which each pickup point is visited exactly once, Desrochers et al. (1992) presented a set covering model which no longer requires the routes in R to be elementary. In the SC model, a_{ir} represents the number of times a pickup point i is visited by route r. It can take any positive integer values, not just binary value. Hence a new constraint (3.41) is proposed to replace constraint (3.38).

$$\sum_{r \in R'} a_{ir} y_r \geq 1, \ \forall i \in N \tag{3.41}$$

The SC model provides a lower bound to the SP model. Any feasible solution to the SP model is also feasible to the SC model. On the other hand, any feasible solution to the SC model may be converted to a feasible solution to the SP model. If a SC model's solution is infeasible to the SP model, this means one or more pickup points are visited more than once. The excessive visits, which make the solution infeasible to the SR model, may be eliminated by simply removing the revisited pickup point in a route and applying a shortcut between the upstream pickup point and the downstream pickup point. Since the cost matrix satisfies the triangle inequality, this conversion would not increase the cost of the solution.

Although the relaxation of (3.38) yields a weaker lower bound than that of the SP model because of the existence of non-elementary routes in *R*, the SC model is still more beneficial than the SP model. First, the SC model is numerically more stable than the SP model especially in the environments involving many customers on a same route (Desrochers et al., 1992). Second, the linear relaxation of the SC model is easier to solve than the SP model. Computation results by Jin et al. (2008) indicate that the SC model improves the speed of column generation.

Since the number of all feasible routes in a CVRPPD instance increases exponentially with the problem size, explicitly enumerating all the feasible routes is not an option for large size CVRPPD. Therefore the column generation based approach is applied to solve the problem. One of the key steps in column generation is to design a sub-problem model for generating columns into *R'* so that *R'* is expanded progressively towards the optimum solution.

Sub-Problem Model

Every linear programing problem has an associated dual linear programing problem. For the CVRPPD, the LRMP is referred to as a primal problem. Let $y_r = \left\{ y_1, y_2, \ldots, y_{|R'|} \right\}$ be the optimal solution to the LRMP. It is necessary to identify whether y_r is also an optimum solution to the LMP.

Let $\delta = \left\{ \delta_1, \delta_2, \ldots, \delta_{|R|} \right\}$ be the set of dual variables associated with (3.41) and $\delta = \left\{ \delta_1, \delta_2, \ldots, \delta_n \right\}$ be the dual optimal solution with respect to y_r. The dual of the linear relaxation of the master problem (LRMPD) is represented as follows.

$$LRMPD: \max \sum_{i \in N} \delta_i \tag{3.42}$$

$$\sum_{i \in N} a_{ir} \delta_i \leq c_r, \ \forall r \in R' \tag{3.43}$$

$$\delta_i \geq 0, \ \forall i \in N \tag{3.44}$$

Clearly δ satisfies constraints (3.43) for all $r \in R'$. Hence if we can prove that δ satisfies constraints (3.43) for all $r \in R$, δ is optimum for the LMPD and thus y_r is optimum for the LMP according to the duality theorem (Boyd et al., 2009). Instead, if there is a route r, $r \in R$ that violates the constraints (3.43), the current δ is not optimum for LMPD. The corresponding route r, which causes the violation, can be added into R' of the LRMP. The LRMP is then solved again. This process repeats until no route violating constraints (3.43) can be found (See Figure 3.3). At this point, the optimum solutions, y and δ, are found for the LMP and LMPD, respectively.

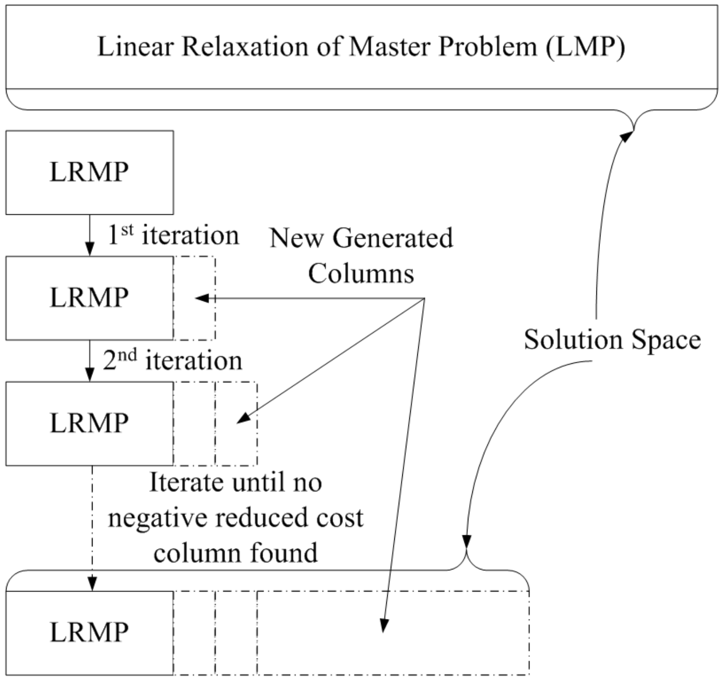

Figure 3.3 Columns Set Augmentation

Figure 3.3 illustrates the relationship between the LRMP and the LMP in terms of the number of columns. The first row of Figure 3.3 shows the complete set of columns. The rest of rows demonstrates how the columns set is augmented towards the optimum solution for each iteration of the column generation process.

Let c_r be the reduced cost of a route *r*. c_r is formulated as follows.

$$c_r = c_r - \sum_{i \in N} a_{ir} \delta_i, \ \forall r \in R \tag{3.45}$$

The sub-problem now is to find a feasible route *r* with negative c_r. The sub-problem must be able to efficiently price out all feasible routes, that is the reason it is usually called pricing problem. Then, the sub-problem decomposes into *n* identical problems, each of which is an elementary shortest path problem with capacity constraint defined on the same graph as the master problem. The ESPPCC model is formulated as follows.

$$\min \sum_{i \in V} \sum_{j \in V \setminus i} c_{ij} z_{ij} \tag{3.46}$$

$$\sum_{i \in M} \sum_{j \in S} z_{ij} = 1 \tag{3.47}$$

$$\sum_{i \in N} \sum_{j \in M} z_{ij} = 1 \tag{3.48}$$

$$\sum_{i \in S} \sum_{j \in N} z_{ij} = 1 \tag{3.49}$$

$$\sum_{i \in V} z_{io} - \sum_{j \in V} z_{oj} = 0, \ \forall o \in V, \ o \neq i, \ o \neq j \tag{3.50}$$

$$z_{ij} = 0, \ \forall i \in S, \forall j \in M \tag{3.51}$$

$$z_{ij} = 0, \ \forall i \in N, \forall j \in S \tag{3.52}$$

$$z_{ij} = 0, \ \forall i \in M, \forall j \in N \bigcup M \tag{3.53}$$

$$u_i - u_j + Q z_{ij} \leq Q - d_j, \ \forall i, j \in N, \ i \neq j, \ \textit{such that } d_i + d_j \leq Q \tag{3.54}$$

$$d_i \leq u_i \leq Q, \ \forall i \in N \tag{3.55}$$

$$z_{ij} \in \{0,1\}, \ \forall i, j \in V \tag{3.56}$$

Where,

c_{ij} = Cost of using arc *(i, j)*, where $c_{ij} = c_{ij} - \frac{\delta_i}{2} - \frac{\delta_j}{2}$

z_{ij} is the decision variable that represents flow in the network.

$$z_{ij} = \begin{cases} 1, \text{ if arc } (i, j) \text{ is used in the shortest path}, \ \forall i \in V, \ \forall j \in V, \ i \neq j \\ 0, \text{ otherwise} \end{cases}$$

The objective is to find the shortest path with negative reduced cost that covers a subset of pickup points N', $N' \subseteq N$. Constraints (3.47) – (3.50) are flow conservation constraints. Constraints (3.51) – (3.53) are connectivity constraints. Constraints (3.54) – (3.55) are the sub-tour elimination constraint where u_i is the vehicle load after visiting pickup point *i*. Constraint (3.56) ensures the integrality.

Since the ESPPCC is NP-hard (Dror, 1994), allowing cycles on the shortest path by relaxing some of the constraints, which changes the ESPPCC to the non-elementary Shortest Path Problem with Capacity Constraint (SPPCC) (Desrosiers et al., 1992; Irnich and Villeneuve, 2006), becomes imperative regarding the computational burden. However, allowing cycles on the shortest path will expand the columns set *R* and thus provide a weaker lower bound to the master problem. Therefore researches focused on compromising between complexity and quality. Beasley and Christofides (1989) imposed a new resource on each node indicating the vertices that has been previously visited so as to prevent cycles. Desrochers et al. (1992) provided a 2-cycle elimination algorithm which eliminates the cycles with *i-j-i* form. Irnich and Villeneuve (2006) extended the 2-cycle elimination to *k*-cycle elimination where cycles containing *k* (or less) nodes are removed.

CDVRPPD Model

In a real-time scheme, a planning horizon [0, *H*] is applied to the evacuation process as illustrated in Figure 3.4, where *H* is the maximum evacuation time. *H* is evenly divided into $\lfloor H/l \rfloor$ intervals with equal length *l*. The length *l* is determined based on the problem size.

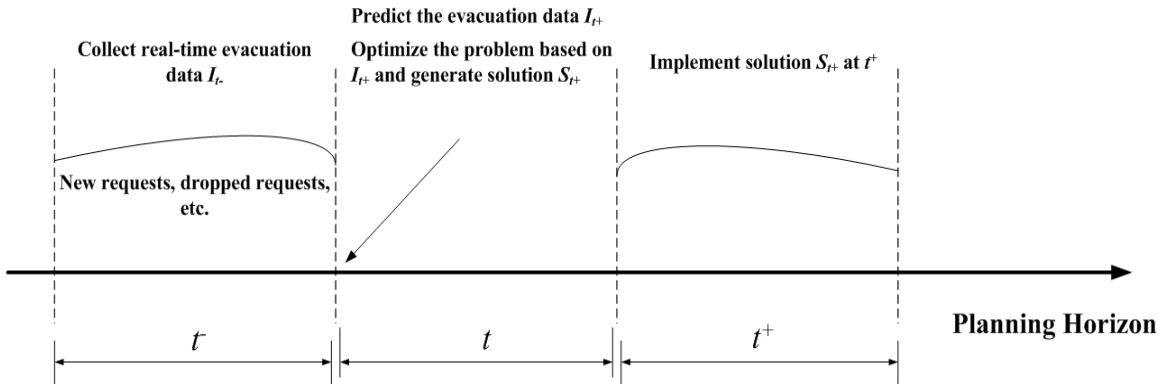

Figure 3.4 Time Intervals over the Planning Horizon

Notations applied to the CDVRPPD model are as follows:

N_t = Set of unfulfilled pickup points in t, $\forall t \in T$.

F_t = Set of fulfilled pickup points in t, $\forall t \in T$.

E_t = Set of new pickup points in t, $\forall t \in T$.

V_t = Set of vertices in t, $V_t = N_t \bigcup S \bigcup M \bigcup L_t, \forall t \in T$.

A_t = Set of arcs in t, $\forall t \in T$.

c_{ijt} = Travel time on arc (i, j) in t, $\forall (i, j) \in A_t, \forall t \in T$.

c_{ijt}' = Predicted travel time on arc (i, j) in t, $\forall (i, j) \in A_t, \forall t \in T$.

c_{rt} = Travel time of a feasible route r in t, $\forall r \in R_t$, $\forall t \in T$.

H = Maximum allowed evacuation time.

l = Interval length.

T = Set of intervals, $T = \left\{t_0, t_1, \ldots, t_{\lfloor H/l \rfloor}\right\}$.

t = Time interval, $t \in T$.

t^- = Preceding interval to t, $\forall t \in T / t_0$.

t^+ = Subsequent interval to t, $\forall t \in T / t_{\lfloor H/l \rfloor}$.

R_t = Set of feasible routes in t, $\forall t \in T$.

R_t' = Subset of feasible routes in t, $\forall t \in T$.

L_t = Set of vehicle locations in t, $\forall t \in T$.

c_{ijt} = Cost of using arc (i, j) in t, $\forall t \in T$, where $c_{ijt} = c_{ijt} - \frac{\delta_{it}}{2} - \frac{\delta_{jt}}{2}$

u_{it} = Vehicle load after visiting pickup point i in t, $\forall i \in N_{t^+} \cup L_{t^+}$, $\forall t \in T$.

As shown in Figure 3.4, at the beginning of interval t, all the evacuation data, including vehicle locations L_t, unfulfilled demands N_t, and arc travel times c_{ijt}, are updated. A CDVRPPD model is then formulated for generating transit vehicle routes applicable in t^+. The CDVRPPD model is proactive that the evacuation data in t^+ are estimated based on the evacuation data in t^-.

Link travel time is predicted by weighted moving average (Hunter, 1986). The weighted moving average method uses a weighting factor which gives more importance to recent observations while not discarding the older observations. The predicted travel time on arc (i, j) in t^+, c'_{ijt^+}, is calculated using Equation (3.57) – (3.58).

$$c'_{ijt^+} = \lambda c_{ijt} + (1-\lambda)\tau_{ijt^-} \tag{3.57}$$

Where, c_{ijt} is the observed travel time on arc (i, j) in t. λ is the weighting factor that $0 < \lambda \leq 1$. τ_{ijt^-} is the average of observed travel times on arc (i, j) in t^-, which is calculated using Equation (3.58).

$$\tau_{ijt^-} = \frac{\sum_{t=t_0}^{t^-} c_{ijt}}{h} \tag{3.58}$$

Where, h is the number of observations recorded in the historical data set.

The robustness and accuracy of the weighted moving average method depends on the value of the weighting factor, λ. λ determines how responsive a forecast is to travel time surge. For a real-time system with short-term travel time forecasting, a value of $\lambda = 0.38$ is suggested by Raiyn and Toledo (2014). The major advantage of the weighted moving average method is that minimizing data storage and computing requirements, which makes it suitable for real-time applications.

The set of vehicle locations in t^+, L_{t^+}, is determined based on the vehicle locations in t, L_t, and the arc travel time c_{ijt}. L_{t^+} is constantly updated over time. It is necessary to include L_{t^+} when formulating the CDVRPPD model. Each vehicle k's location is usually considered as a depot where the vehicle k departs in t^+. Thus the problem turns to be a CDVRPPD with $|L_{t^+}|+1$ depots, which consists of $|L_{t^+}|$ temporary depots at the vehicle locations and one real depot.

For the vehicle routing problem with multiple depots, one of the most common methods is clustering which assigns pickup points to a depot. This procedure is deemed as a Generalized Assignment Problem (GAP). Once the GAP is solved, the problem is decomposed into multiple single-depot problems.

For the CDVRPPD with multiple depots, vehicle k's temporary location $L_{t^+}^k$ is counted as a depot, however, no vehicle other than vehicle k can start from L_t^k. In this particular case, the problem can be converted to a CDVRPPD with single depot by introducing dummy pickup points in the network. A dummy pickup point n_k is added at vehicle k's location. Demand of n_k equals to vehicle k's load. Service time at n_k is 0. Travel time from n_k to other nodes is calculated according to their distance. In particular, travel time from depot s_0 to n_k is set to 0 and travel times from pickup points and shelters to n_k are set to infinite. After adding $\left|L_{t^+}\right|$ dummy pickup points, all the temporary depots are replaced by dummy pickup points. The problem is reduced to a CDVRPPD with single depot. The cost of adding $\left|L_{t^+}\right|$ dummy pickup points is that $\left|L_{t^+}\right|$ rows are added into the model. However, the complexity of the model is greatly reduced.

The set of unfulfilled pickup points in t, N_t, is formulated by Equation (3.59).

$$N_t = \begin{cases} N, \text{ when } t = t_0 \\ N_{t^-} \cup E_{t^-} \setminus F_{t^-}, \text{ otherwise} \end{cases} \tag{3.59}$$

The set of unfulfilled pickup points in t^+, N_{t^+}, is predicted by Equation (3.60). When estimating N_{t^+}, it does not take into account the pickup requests that arrive in t. The pickup requests that arrive in t will be considered in the next interval. The total pickup points set in t^+, including both the real pickup points and the dummy pickup points, is $N_{t^+} \cup L_{t^+}$.

$$N_{t^+} = N_t \setminus F_t, \ \forall t, t^+ \in T \tag{3.60}$$

In order to explicitly describe the graph G in the dynamic environment, two sets of vertices are introduced when representing the network in t. Given a vertex i, $V_{it^+}^{+}$ is defined as the set of vertices j such that arc (i, j) is not prohibited in t^+, i.e., the vertices in the set of $V_{it^+}^+$ are directly reachable from i. Similarly, $V_{it^+}^-$ denotes the set of vertices j from which vertex i is directly reachable in t^+.

$$V_{it^+}^+ = \begin{cases} N_{t^+} \cup L_{t^+}, \text{ if } i \in S \\ N_t \cup M \setminus \{i\}, \text{ if } i \in N_{t^+} \cup L_{t^+} \\ S, \text{ if } i \in M \end{cases}$$

$$V_{it^+}^- = \begin{cases} M, \text{ if } i \in S \\ S \cup N_{t^+} \setminus \{i\}, \text{ if } i \in N_{t^+} \\ S, \text{ if } i \in L_{t^+} \\ N_t \cup L_{t^+}, \text{ if } i \in M \end{cases}$$

Let γ_{irt^+} be a variable indicating if route r visits pickup point i in t^+.

$$\gamma_{irt^+} = \begin{cases} 1, \text{ if route } r \text{ visits pickup point } i \text{ in } t^+, \ \forall t^+ \in T, \ \forall r \in R_{t^+}', \ \forall i \in N_{t^+} \cup L_{t^+} \\ 0, \text{ otherwise} \end{cases}$$

The CDVRPPD model has a binary variable x_{rt^+} indicating whether route r is used in t^+.

$$x_{rt^+} = \begin{cases} 1, \text{ if route } r \text{ is used in } t^+, \ \forall t^+ \in T, \ \forall r \in R_{t^+}' \\ 0, \text{ otherwise} \end{cases}$$

For interval t^+, the master problem model of CDVRPPD is formulated by (3.61) – (3.65).

$$\min \sum_{r \in R_{t^+}'} c_{rt^+} x_{rt^+} \tag{3.61}$$

$$\sum_{r \in R_{t^+}'} \gamma_{irt^+} x_{rt^+} \geq 1, \ \forall i \in N_{t^+} \cup L_{t^+} \tag{3.62}$$

$$x_{rt^+} \in \{0,1\},\ \forall r \in R_{t^+}',\ \forall t^+ \in T \tag{3.63}$$

$$R_{t^+}' \subseteq R_{t^+} \tag{3.64}$$

$$\forall t^+ \in T \tag{3.65}$$

The master problem model is to select transit vehicle routes over the planning horizon to fulfill pickup requests from registered evacuees and unregistered evacuees so as to minimize the total travel cost. Constraints (3.62) ensure that each pickup point i is covered by at least one route r in the routes set R_{t^+}. It also requires a sub-problem model to generate routes with negative reduced cost. Let $\delta_{t^+} = \left\{\delta_{1t^+}, \delta_{2t^+}, \ldots, \delta_{|N_{t^+} \cup L_{t^+}|}\right\}$ be the set of dual variables associated with (3.62) and $\delta_{t^+} = \left\{\delta_{1t^+}, \delta_{2t^+}, \ldots, \delta_{|N_{t^+} \cup L_{t^+}|}\right\}$ be the dual optimal solution. The sub-problem model is formulated as follows.

$$\min \sum_{i \in V_{t^+}} \sum_{j \in V_{t^+}} c_{ijt^+} z_{ijt^+} \tag{3.66}$$

$$\sum_{i \in M} \sum_{j \in S} z_{ijt^+} = 1 \tag{3.67}$$

$$\sum_{i \in V_{jt^+}^-} \sum_{j \in M} z_{ijt^+} = 1 \tag{3.68}$$

$$\sum_{i \in S} \sum_{j \in V_{it^+}^+} z_{ijt^+} = 1 \tag{3.69}$$

$$\sum_{i \in V_{t^+}} z_{iot^+} - \sum_{j \in V_{t^+}} z_{ojt^+} = 0,\ \forall o \in V_{t^+},\ o \neq i,\ o \neq j \tag{3.70}$$

$$z_{ijt^+} = 0,\ \forall i \in S,\ \forall j \in M \cup S \tag{3.71}$$

$$z_{ijt^+} = 0,\ \forall i \in N_{t^+} \cup L_{t^+},\ \forall j \in S \tag{3.72}$$

$$z_{ijt^+} = 0,\ \forall i \in M,\ \forall j \in N_{t^+} \cup L_{t^+} \cup M \tag{3.73}$$

$$u_{it^+} - u_{jt^+} + Qz_{ijt^+} \le Q - d_j,\ \forall i, j \in N_{t^+} \cup L_{t^+},\ i \ne j,\ such\ that\ d_i + d_j \le Q \tag{3.74}$$

$$d_i \le u_{it^+} \le Q,\ \forall i \in N_{t^+} \cup L_{t^+} \tag{3.75}$$

$$z_{ijt^+} \in \{0,1\},\ \forall i, j \in V_{t^+} \tag{3.76}$$

z_{ijt^+} is decision variable.

$$z_{ijt^+} = \begin{cases} 1,\ \text{if arc } (i,j) \text{ is used in the shortest path in } t^+, \\ \forall t^+ \in T,\ \forall i \in V_{t^+},\ \forall j \in V_{t^+},\ i \ne j \\ 0,\ \text{otherwise} \end{cases}$$

The CDVRPPD model is similar to the CVRPPD model. Both of them are formulated based on a master problem model and sub-problem model structure. Constraints (3.67) – (3.70) are flow conservation constraints. Constraints (3.71) – (3.73) are connectivity constraints. Constraints (3.74) – (3.75) are the sub-tour elimination constraint. Constraints (3.76) ensure the integrality.

Model Variants

Dynamic Interval;

The interval length, *l*, in which the optimization process is performed, is directly related with the network size. It is an important parameter in the CDVRPPD model development. When a new pickup request is collected in t^-, it will be processed in *t* and then an updated routing plan will be implemented in t^+. Hence a new pickup request has to wait at least an interval until an updated routing plan is implemented. On one hand, a short interval is beneficial to decreasing the waiting time of the new pickup request; on the other hand, a long interval is imperative at the initial stage of the planning horizon due to the computational burden. In some instances that new pickup requests from unregistered evacuees are infrequent, the network size will decrease after the first several intervals. As a result, the computational burden will be reduced. For the CDVRPPD, it is necessary to adjust the interval length dynamically in order to keep the model reacting to the evacuation data updates.

In order to overcome the deficiency of fixed-length interval, dynamic interval is implemented. The length of interval t is calculated based on the computation time in t^-.

$$l_t = t_{ct^-}\beta,\ t, t^- \in T,\ \beta > 1 \tag{3.77}$$

Where, l_t is the length of interval t. t_{ct^-} is the computation time in t^-. β is the incremental factor which represents the percent of increase. At the initial stage of evacuation process, the network size increases with the new pickup request coming into the system. In response, the interval length will increase accordingly by multiplying the incremental factor β. It is expected that the computation time in t shows downtrend when the number of completed requests in t exceeds the number of new requests in t^-. In this case, the incremental factor β makes the interval length falling lags behind the computation time. It ensures a surplus of time each interval which could be used to deal with uncertainties.

Multiple Depots CDVRPPD;

In some cases, the transit vehicles are not placed in a single depot. Instead, they are pre-allocated to multiple depots. An optimized allocation could make the emergency evacuation more efficient. This operation actually changes the CDVRPPD into a multi-depot CDVRPPD (MCDVRPPD), which obviously increases difficulty. In this section, a generic MCDVRPPD model is proposed in the context of emergency evacuation.

A dummy base depot s_0 where all routes start and end is introduced. The introduction of the base depot could effectively convert the MCDVRPPD into the CDVRPPD. The travel costs between the base depot and the other nodes in the network are described as follows. All other travel costs are set as defined by the original CDVRPPD.

1. Travel costs from the base depot to the other depots are set to 0.
2. Travel costs from the base depot to the pickup points are set to infinite.
3. Travel costs from the base depot to the dummy pickup points are set to infinite.
4. Travel costs from the base depot to the shelters are set to infinite.
5. Travel costs from the pickup points to the base depot are set to infinite.
6. Travel costs from the dummy pickup points to the base depot are set to infinite.
7. Travel costs from the other depots to the base depot are set to infinite.
8. Travel costs from the shelters to the base depot are set to 0.

A solution to the MCDVRPPD includes a set of routes that originate from the base depot, then perform a number of pickups, and finally end at a shelter. Each route that starts from the base depot must pass a real depot before pickup. The two vertices set, $V_{it^+}^{+}$ and $V_{it^+}^{-}$, are also updated as follows after the introduction of the base depot.

$$V_{it^+}^{+}=\begin{cases} S \setminus s_0, \text{ if } i=s_0 \\ N_{t^+} \bigcup L_{t^+}, \text{ if } i \in S \setminus s_0 \\ N_t \bigcup M \setminus \{i\}, \text{ if } i \in N_{t^+} \bigcup L_{t^+} \\ s_0, \text{ if } i \in M \end{cases}$$

$$V_{it^+}^{-}=\begin{cases} M, \text{ if } i=s_0 \\ s_0, \text{ if } i \in S \setminus s_0 \\ S \bigcup N_{t^+} / \{i, s_0\}, \text{ if } i \in N_{t^+} \\ S, \text{ if } i \in L_{t^+} \\ N_t \bigcup L_{t^+}, \text{ if } i \in M \end{cases}$$

The MCDVRPPD model is developed based on a set-covering formulation. The master problem model is same as (3.61) – (3.65). The sub-problem model is presented by (3.78) – (3.91).

$$\min \sum_{i \in V_{t^+}} \sum_{j \in V_{t^+}} c_{ijt^+} z_{ijt^+} \tag{3.78}$$

$$\sum_{i \in M} z_{ijt^+} = 1, \; j = s_0 \tag{3.79}$$

$$\sum_{j \in S \setminus s_0} z_{ijt^+} = 1, \; i = s_0 \tag{3.80}$$

$$\sum_{i \in S \setminus s_0} \sum_{j \in V_{it^+}^{+}} z_{ijt^+} = 1 \tag{3.81}$$

$$\sum_{i \in V_{jt^+}^{-}} \sum_{j \in M} z_{ijt^+} = 1 \tag{3.82}$$

$$\sum_{i \in V_{t^+}} z_{iot^+} - \sum_{j \in V_{t^+}} z_{ojt^+} = 0, \; \forall o \in V_{t^+}, \; o \neq i, \; o \neq j \tag{3.83}$$

$$z_{ijt^+} = 0, \; \forall i \in M, \; \forall j \in V_{t^+} \setminus s_0 \tag{3.84}$$

$$z_{ijt^+} = 0,\ i = s_0,\ \forall j \in N_{t^+} \cup L_{t^+} \cup M \tag{3.85}$$

$$z_{ijt^+} = 0,\ \forall i \in S \setminus s_0,\ \forall j \in M \cup S \tag{3.86}$$

$$z_{ijt^+} = 0,\ j = s_0,\ \forall i \in V_{t^+} \setminus M \tag{3.87}$$

$$z_{ijt^+} = 0,\ j = S \setminus s_0,\ \forall i \in V_{t^+} \setminus s_0 \tag{3.88}$$

$$u_{it^+} - u_{jt^+} + Qz_{ijt^+} \le Q - d_j,\ \forall i, j \in N_{t^+} \cup L_{t^+},\ i \ne j,\ such\ that\ d_i + d_j \le Q \tag{3.89}$$

$$d_i \le u_{it^+} \le Q,\ \forall i \in N_{t^+} \cup L_{t^+} \tag{3.90}$$

$$z_{ijt^+} \in \{0,1\},\ \forall i, j \in V_{t^+} \tag{3.91}$$

(3.78) is the objective function. Constraints (3.79) – (3.83) are flow conservation constraints. Constraints (3.85) – (3.88) are connectivity constraints. Constraints (3.89) – (3.90) are the sub-tour elimination constraint. Constraints (3.91) ensure the integrality.

CHAPTER IV

SOLUTION ALGORITHMS AND SYSTEM DESIGN

In this chapter, solution algorithms to the CDVRPPD model are presented. The computational results on benchmark problems are compared to other studies in the literature. A SmartEvac system is developed to manage the transit vehicle routing in emergency evacuation. Models and algorithms discussed in Chapter III and Chapter IV are implemented in the SmartEvac system.

Solution Algorithms

In order to solve the CDVRPPD without enumerating all the routes, a column generation approach is applied to the problem. The general process of the column generation approach is presented as follows. First, an initial subset R_{t^+}' of all feasible routes R_{t^+} is enumerated. The LRMP, whose routes set is restricted to R_{t^+}', is then solved and the dual solution is obtained. The dual solution is utilized in a sub-problem to determine if there are any routes that should be added to R_{t^+}' towards an optimum. If new routes are found and added to R_{t^+}', the LRMP is then resolved with respect to the expanded R_{t^+}'. This process repeats until no additional routes can be found that further optimize the objective. At this point, the optimum solution to the LMP with R_{t^+} is found by solving the LRMP with R_{t^+}'. The optimum solution to the LMP is not necessarily integer. Actually, most of time it is fractional. If it is fractional, the final step is to solve the RMP as an integer problem in order to get an integer solution. A flow chart of the column generation method is shown in Figure 4.1.

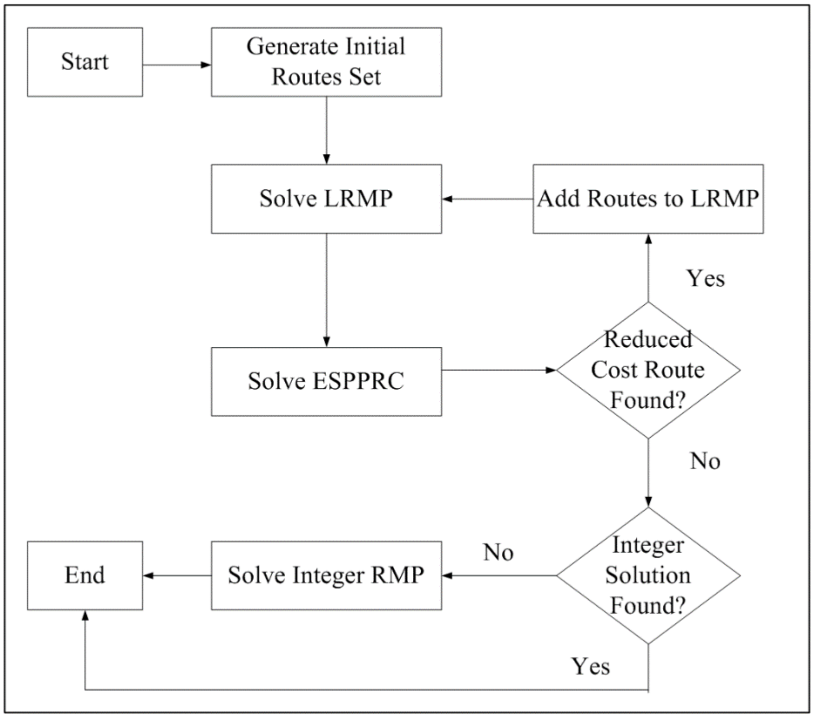

Figure 4.1 Column Generation Approach Flow Chart

The specific procedures of the column generation method are presented by the following steps.

Step 1. Create an initial subset of columns, R_{t^+}', $R_{t^+}' \subseteq R_{t^+}$.

Step 2. Solve the LRMP and get the optimal solution y_{rt^+}, $r \in R_{t^+}'$ and the corresponding dual solution δ_{t^+}.

Step 3. Solve the ESPPCC sub-problem with δ_{t^+}. Identify routes $r,\ r \in R_{t^+}$

satisfying $c_{rt^+} < 0$.

Step 4. If $r \neq \varnothing$, add r into R_{t^+}' and go to step 2.

Step 5. If $r \neq \varnothing$, check if y_{rt^+} is an integer solution.

Step 6. If y_{rt^+} is integer, go to step 8.

Step 7. If y_{rt^+} is fractional, solve the integer RMP.

Step 8. End.

Initializing Set of Columns

Firstly, a set of columns is initialized for the LRMP. The initial set of columns needs to include at least a feasible solution to the LRMP. A common initial set is made of routes visiting a single pickup point, i.e. routes of type $C-N-M-C$. Since a good set of initial routes helps to generate routes with low reduced cost (Toth et al., 2001), quick heuristics are implemented to generate the initial routes set with high quality.

In t_0, the first interval of the planning horizon, the Clarke and Wright Savings Algorithm (Clarke and Wright, 1964) is applied to create initial routes. The Clarke and Wright Savings Algorithm is based on notion of savings. The basic idea is that a cost saving $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ is generated when two routes $(0,\cdots,i,0)$ and $(0,j,\cdots,0)$ can be feasibly merged in to a single route $(0,\cdots,i,j,\cdots,0)$. The flow chart of the Clarke and Wright Savings Algorithm is shown in Figure 4.2.

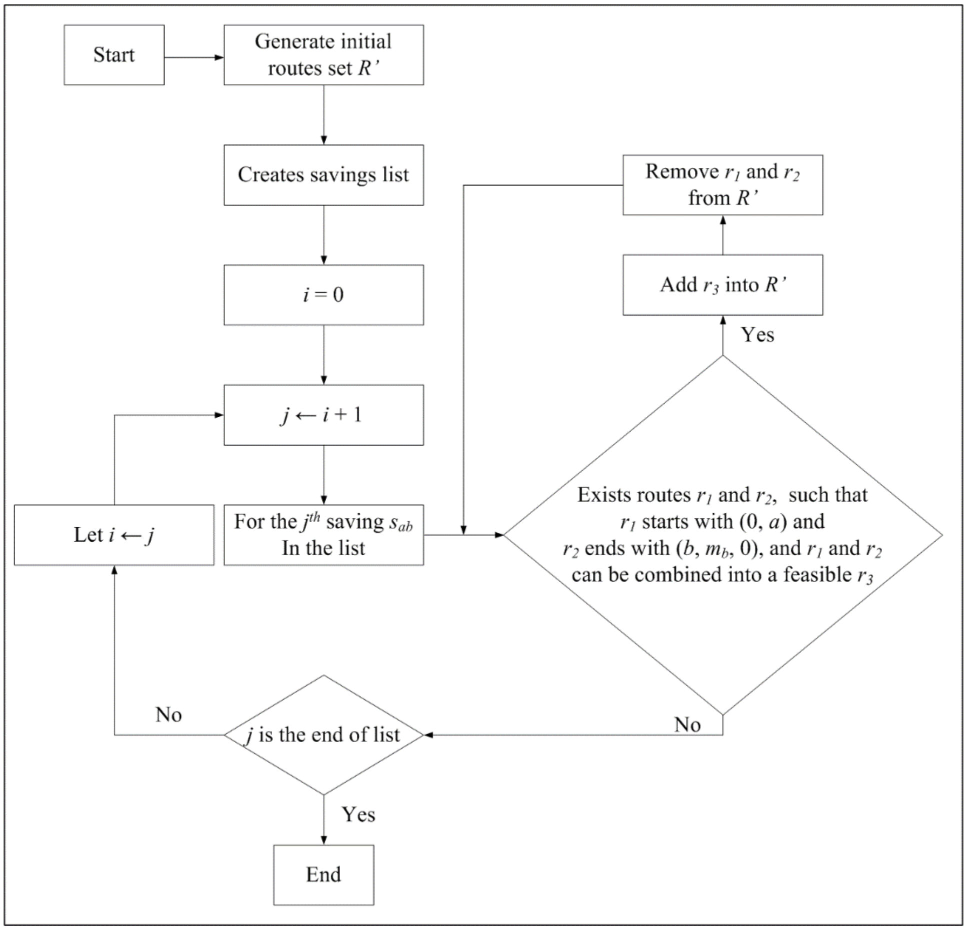

Figure 4.2 Clarke-Wright Savings Algorithm Flow Chart

The specific procedures of the algorithm are implemented as follows.

Step 1. Create an initial routes set *R'* including |*N*| vehicle routes. Each route has the following route structure, $(0, i, m_i, 0)$, $i \in N$, where m_i is the nearest shelter to pickup point *i*.

Step 2. Calculate the cost savings $s_{ij} = c_{im_i} + c_{m_i 0} + c_{0j} - c_{ij}, \ \forall i, j \in N, \ i \neq j$, where $c_{m_i 0} = 0, \ \forall i \in N$. Rank the savings s_{ij} and list them in descending order. This creates the savings list.

Step 3. Process the savings list beginning with the topmost entry. For s_{ij}, find route r_1, $r_1 \in R'$ that starts with $(0, j)$ and route r_2, $r_2 \in R'$ that ends with $(i, m_i, 0)$. Combine r_1 and r_2 into a new route r_3 by deleting $(0, j)$ and $(i, m_i, 0)$ and introducing (i, j). If r_3 is feasible to the model, add r_3 into R' and remove r_1 and r_2 from R'.

Step 4. Iterate to the next entry in the savings list until the end.

The advantage of the Clarke and Wright Savings Algorithm lies in its simplicity and speed, which makes it suitable to generate a good set of initial routes. It typically runs within 0.5 seconds on Christofides, Mingozzi and Toth's (1979) benchmark instances with 100 nodes.

The initialization step is handled differently for interval $t^+ \subseteq T \setminus \{t_0\}$. The initial routes set R_{t^+}' in t^+ is created based on the routes set R_t in t. First, the Clarke and Wright Savings Algorithm is used to generate an initial routes set R_{ini} that serves the new pickup points in E_t^-. Second, the routes set R_t is updated. The vehicle routes that have been completed in t are removed from R_t. The pickup points that have been visited in t are removed from the routes as well. Third, an insertion algorithm is applied to R_t. For a new pickup point n in E_t and a route r in R_t, the algorithm inserts the new pickup point n to an arc (i, j) in r such that the incremental cost of inserting n between i and j is minimal. The flow chart of the insertion algorithm is shown in Figure 4.3.

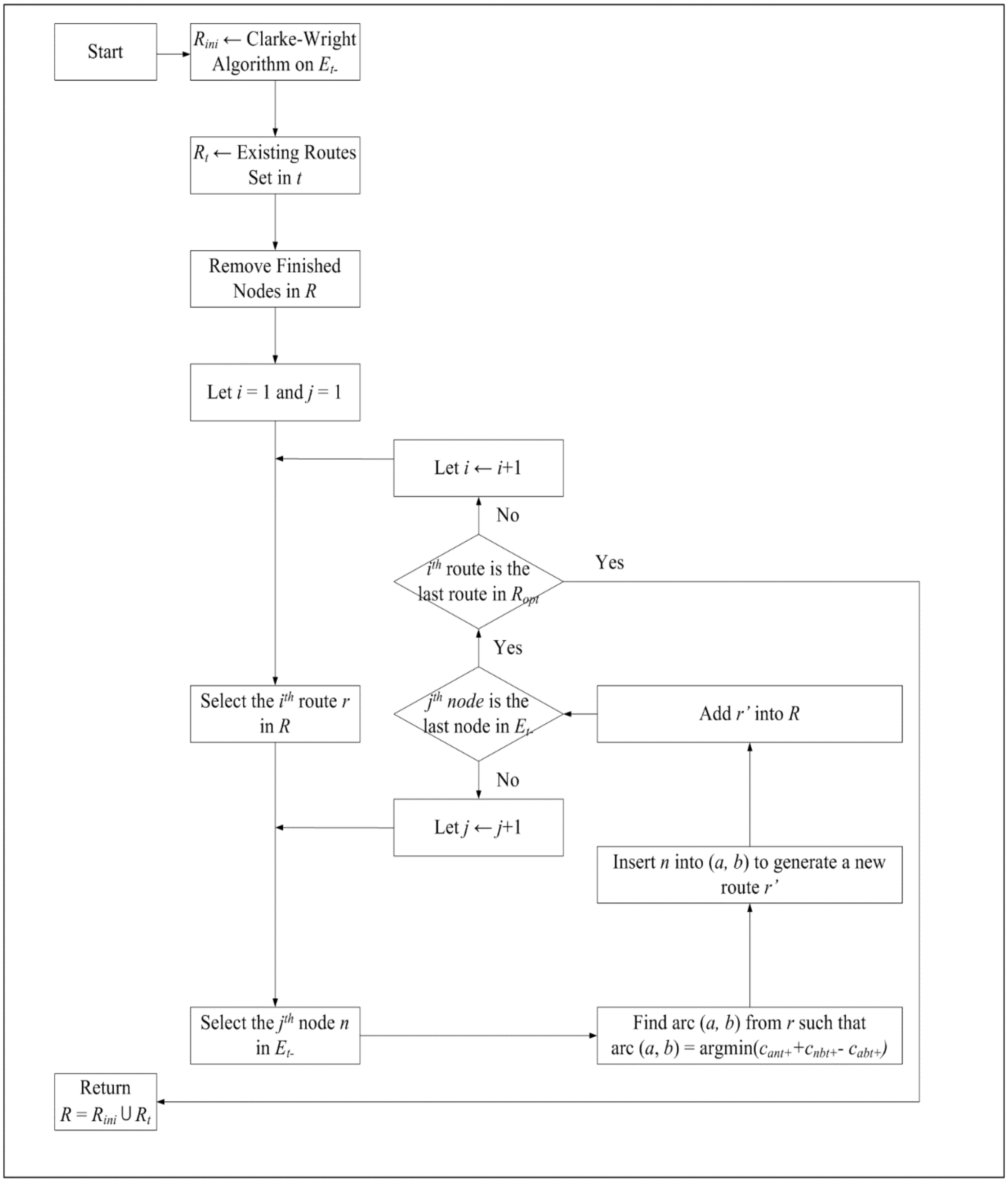

Figure 4.3 Insertion Algorithm Flow Chart

The specific procedures are described as follows.

For every pickup point n in E_t

For every route r in R_t

Find an arc (*i, j*) in *r* such that $c_{int^+} + c_{njt^+} - c_{ijt^+}$ is minimal.

$$\arg\min_{(i,j)} \left(c_{int^+} + c_{njt^+} - c_{ijt^+} \right)$$

Construct a new route *r'* by replacing (*i, j*) with (*i, n, j*).

If *r'* is feasible to the model, add *r'* into R_t.

Update R_t

End

Cycle Elimination Algorithm for the ESPPCC

The objective of the pricing sub-problem is to identify the routes with negative reduced cost. The first step is to find the shortest path to each pickup point. This step is considered to be |*N*| ESPPCCs, each of which is NP-hard (Dror, 1994). For each ESPPCC, the task is to find the shortest partial path *r* from Node 0 to Node $i,\ i \in N$. Since shelters are not involved in finding the shortest partial paths, the network can be simplified by removing the shelters. Solving the ESPPCC is the most time consuming procedure in the column generation and thus significantly affects the performance of the optimization. Algorithms solving the ESPPCC in the literature include dynamic programming, branch-cut, and classic heuristics.

In this section, a cycle elimination algorithm is proposed based on standard labeling techniques presented by Desrochers (1988), Beasley and Christofides (1989), and Feillet et al. (2004). The CE algorithm first turns the ESPPCC to 2-cycle SPPCC by allowing cycles with length ≤ 2. Then a resource constraint is iteratively imposed upon the model to eliminate cycles with length > 2. Resource in the ESPPCC is related to capacity, time, and node availability etc., whose consumption is always nonnegative. The fundamental of the CE algorithm is based on Desrochers' (1988) labeling algorithm which associates each potential partial path with a label indicating the consumption of resources.

The CE algorithm creates labels for each node i, $i \in N$. Each label l_i represents a partial path X_i from node 0 to node i. l_i includes a pointer $Pre(l_i)$ which links to l_i's parent label. l_i's parent label is defined as the label from which l_i is generated. Let $q(l_i)$ denote the capacity consumed on path X_i and $c(l_i)$ denote the travel cost associated with the path X_i. Thus a label l_i is represented as $l_i(Pre, q, c)$. The algorithm repeatedly extends each label to its successors until all labels have been extended in all feasible ways. The extension is operated by appending an arc (i, j) to path X_i to generate a new path X_j. When a label l_i is extended to a label l_j, the capacity consumption and the path cost are updated as follows,

$$q(l_j) = q(l_i) + d_j \tag{4.1}$$

$$c(l_j) = c(l_i) + c_{ij} \tag{4.2}$$

A new label $l_j(pre, q, c)$ is generated only if,

$$q(l_j) < Q \tag{4.3}$$

It is noted that (4.1) is strictly non-decreasing since $d_j > 0$ for all $j \in N$. The extension of a label l_i is denoted by $Ext(l_i)$.

Dominance Rule

The efficiency of the CE algorithm highly depends on the number of labels generated. Since the extension operation creates exponential number of labels, it is necessary to discard the labels which will not lead to an optimal solution. To this purpose a dominance rule is applied in the label extension so that the algorithm records only non-dominated labels.

If there are two labels $l_{i(1)}$ and $l_{i(2)}$ associated with node i satisfying $q(l_{i(1)}) \leq q(l_{i(2)})$, $c(l_{i(1)}) \leq c(l_{i(2)})$, and $l_{i(1)} \neq l_{i(2)}$, then any feasible extension from label $l_{i(2)}$ will be also feasible from label $l_{i(1)}$. In addition, new labels created based on label $l_{i(1)}$ will always be better than the labels created based on label $l_{i(2)}$, in terms of travel cost (if the objective is to minimize travel cost). Hence the label $l_{i(2)}$ can be discarded. The dominance rule is defined that $l_{i(1)}$ dominates $l_{i(2)}$, denoted by $l_{i(1)} \prec_{dom} l_{i(2)}$, if and only if the following conditions are met.

$$q\left(l_{i(1)}\right) \leq q\left(l_{i(2)}\right) \tag{4.4}$$

$$c\left(l_{i(1)}\right) \leq c\left(l_{i(2)}\right) \tag{4.5}$$

$$l_{i(1)} \neq l_{i(2)} \tag{4.6}$$

Figure 4.4 illustrates the dominance rule that any label in the shaded area will dominate label $l_{i(2)}$.

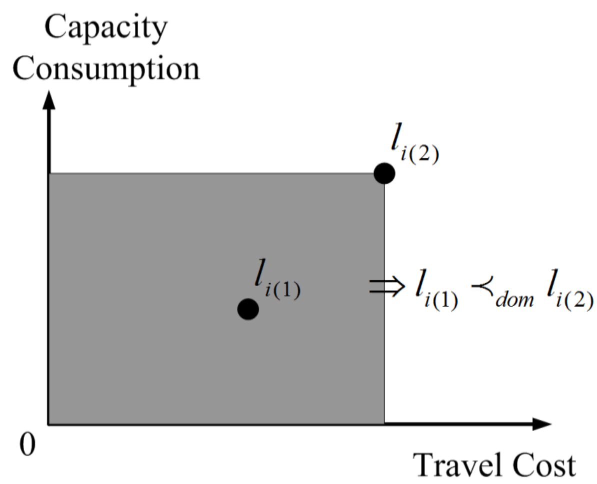

Figure 4.4 Illustration of Dominance Rule

After a new label l_i is generated, it is necessary to check whether the new label is dominated by other labels associated with the same node, and whether the new label dominates other labels. The procedure of dominance check to l_i is denoted by $Dom(l_i)$. Any label which has been identified as being dominated by other labels will be discarded because any extension from the dominated label will be worse than the extension from the dominant label.

Enhanced Dominance Rules for 2-Cycle Elimination

The above dominance rule is applicable in the context of finding a non-elementary shortest path. Because of the existence of negative cost arcs, the relaxation of elementary constraint results in a lot of paths with cycles. This typically weakens the lower bound which leads to a bigger branch-and-bound tree. To improve the lower bound, Houck et al. (1980) proposed an algorithm for solving the SPPRC with 2-cycle elimination. Larsen (1999) extended Houck's algorithm with new definition of labels. In this section, Larsen's method is enhanced by improving the dominance rules.

Let (*i, q*) denote a state of node *i*, which indicates the capacity consumption. For each state, the algorithm generates two types of labels as follows. A new parameter *Typ* is appended to l_i. $Typ(l_j)$ denotes the type of l_i.

1. Strong-dominant label that $Typ(l_j) = Strong$. A strong-dominant label is the prevailing label which dominates the extension. However, a strong-dominant label l_i cannot be extended to its predecessor node. Let $v(l_i)$ denote the associated node of l_i. l_i 's predecessor node is the node which l_i's parent label is associated with, denoted by $v(Pre(l_i))$.
2. Weak-dominant label that $Typ(l_j) = Weak$. A weak-dominant label is dominated by the strong-dominant label. A weak-dominant label has the potential of being extended to the strong-dominant label's predecessor node. It actually provides an alternative path when the extension of the strong-dominant label forms a 2-cycle.

The algorithm can effectively eliminate 2-cycle by introducing a weak-dominant label for each state. As a result, the total number of labels is doubled. Therefore the computational complexity remains the same.

Strong-dominant label and weak-dominant label have different extension rules. When a label l_i is extended to generate a label l_j, the following extension rules are applied.

1. If $Typ(l_i) = Strong$, (4.1) – (4.3) are applied. l_i is not permitted to extend to $v(l_j)$ if $v(pre(l_i)) = v(l_j)$. When $v(pre(l_i)) = v(l_j)$, the weak-dominant label is extended to $v(l_j)$ instead of l_i and (4.1) – (4.3) are applied.
2. If $Typ(l_i) = Weak$, l_i is extendable on the condition that $v(l_j)$ is the predecessor node of the strong-dominant label which dominates l_i, otherwise, l_i is not extendable. When $v(l_j)$ is the predecessor node of the strong-dominant label, l_i is extended instead of the strong-dominant label and (4.1) – (4.3) are applied.

In summary, a strong-dominant label is extendable to any node except its predecessor node. A weak-dominant label is not extendable to any nodes other than the predecessor node of the strong-dominant label. In addition, any extension has to satisfy (4.1) – (4.3).

New dominance rules are added in addition to (4.4) – (4.6), which are described as follows. Assume that $l_{i(1)}$ is an old label at node i and $l_{i(2)}$ is a new generated label at node i. If $l_{i(1)} \prec_{dom} l_{i(2)}$ according to (4.4) – (4.6), $l_{i(2)}$ can be discarded only if one of the following conditions are satisfied.

1. $Typ(l_{i(1)})$ = Strong and $v(Pre(l_{i(1)})) = v(Pre(l_{i(2)}))$.
2. $Typ(l_{i(1)})$ = Weak.

When $Typ(l_{i(1)})$ = Strong and $v(Pre(l_{i(1)})) \neq v(Pre(l_{i(2)}))$, the new generated label $l_{i(2)}$ will proceed to compare with the weak-dominant label dominated by $l_{i(1)}$, to determine whether it can replace the weak-dominant label.

If $l_{i(2)} \prec_{dom} l_{i(1)}$, then $l_{i(1)}$ can be discarded only if one of the following conditions are satisfied.

3. $Typ(l_{i(1)})$ = Strong and $v(Pre(l_{i(1)})) = v(Pre(l_{i(2)}))$.

4. $Typ(l_{i(1)})$ = Weak.

Similarly, when $Typ(l_{i(1)})$ = Strong and $v(Pre(l_{i(1)})) \neq v(Pre(l_{i(2)}))$, the old label $l_{i(1)}$ will become the weak-dominant label which is dominated by $l_{i(2)}$.

Pseudo Code for 2-Cycle Elimination

The pseudo code of the algorithm is presented is this section. The following symbols are used in the code. Γ represents the set of labels which have not been extended. Only strong-dominant labels are placed in Γ. Labels in Γ are placed in lexicographical order. Given two labels $l_i(Pre, q, c, Typ)$ and $l_j(Pre, q, c, Typ)$, l_i is lexicographically smaller than l_j if $q(l_i) < q(l_j)$. $Ext(l_i)$ is the extension procedure which extends label l_i to its successors. The capacity constraint is checked and only feasible labels are produced. $Dom(l_i)$ is the procedure which applies the dominance rule to the new generated label. When a new label $l_i(Pre, q, c, Typ)$ is generated at node i, the dominance rule is applied to check whether the new label is dominated by the old label associated with state $(i, q(l_i))$. Then the strong-dominant label and the weak-dominant label associated with state $(i, q(l_i))$ are updated according to the results of the dominance check. A flow chart of the 2-cycle elimination algorithm is shown in Figure 4.5.

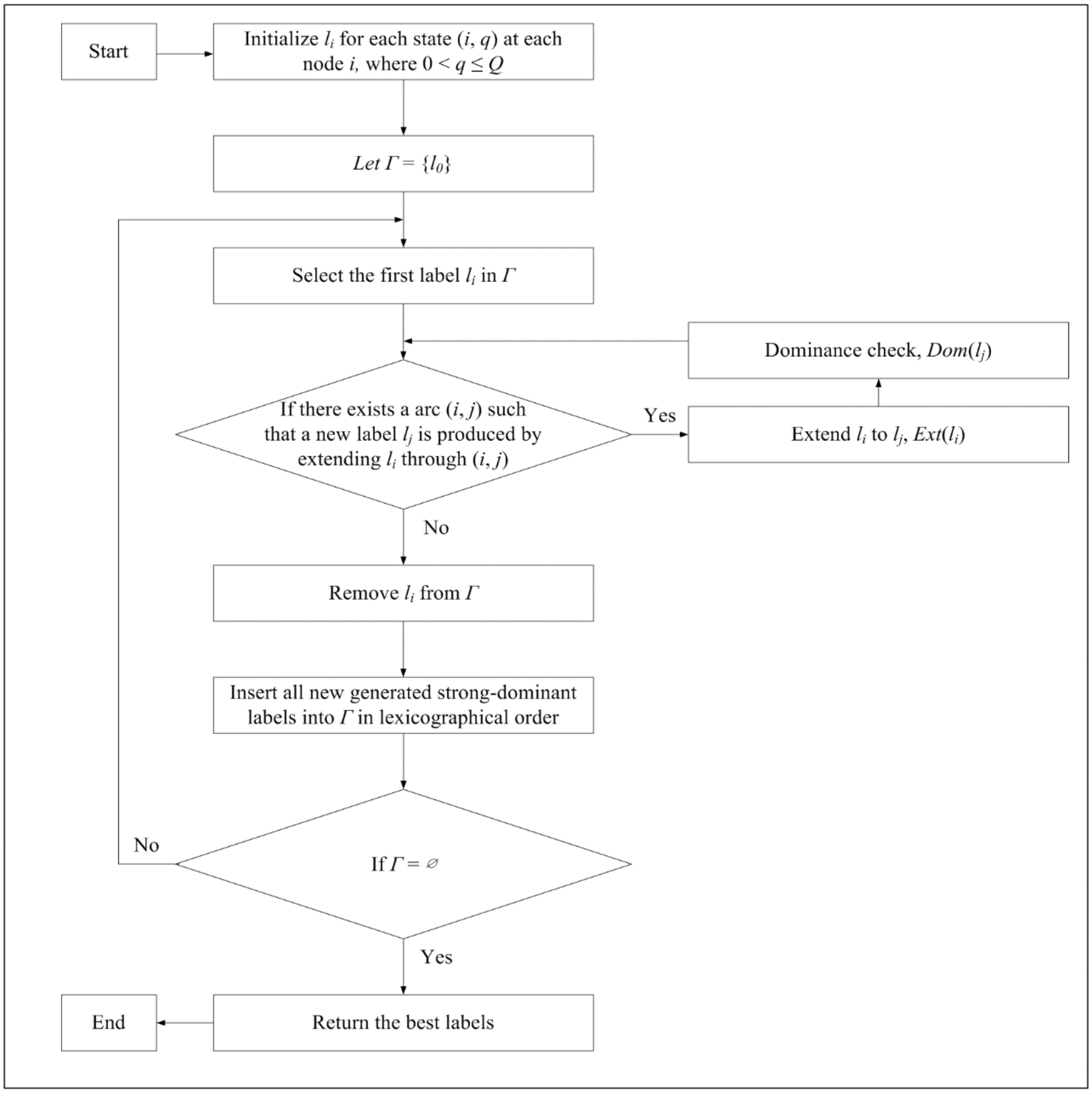

Figure 4.5 2-Cycle Elimination Flow Chart

The specific procedures of the 2-cycle elimination algorithm are presented as follows.

Step 1. Initialization

Initialize the label l_0 = (*Null*, 0, 0, *Strong*) for node 0. Initialize l_i = (*Null*, q, $+\infty$, *Strong*) and l_i' = (*Null*, q, $+\infty$, *Weak*) for all other node i, $i \in N$ and q, $0 < q \leq Q$. Then, let $\Gamma = \{l_0\}$.

Step 2. Label Selection

If $\Gamma = \emptyset$, go to Step 4.

Else, select the first label l_i in Γ. Then Remove l_i from Γ.

Step 3. Label Extension

For all $(i, j) \in A, j \neq 0$

Create a new label $l_j \leftarrow Ext(l_i)$.

Apply the dominance rule to l_j, $Dom(l_j)$.

Step 4. Insert all new generated strong-dominant labels into Γ in lexicographical order. If $\Gamma \neq \emptyset$, go to Step 2.

Step 5. Stop. All labels are extended in all feasible ways.

The algorithm generates a set of strong-dominant labels at each node i, $i \in N$. Then the best label at node i, which indicates a shortest path from node 0 to node i, is found.

A sample network is presented in Figure 4.6. The network consists of seven nodes (0, 1, 2, 3, 4, 5, 6) where node 0 is depot and node 1 to node 6 are customers. The demand of each customer is set to one. The vehicle's capacity is five. The travel cost is marked on the arc. Assume that the travel cost on arc (i, j) equals to the travel cost on arc (j, i), which essentially makes it an undirected graph. It is obvious that there exist three negative cost cycles in the graph, which are 2-3-5, 3-4-6, and 2-3-4-6-5.

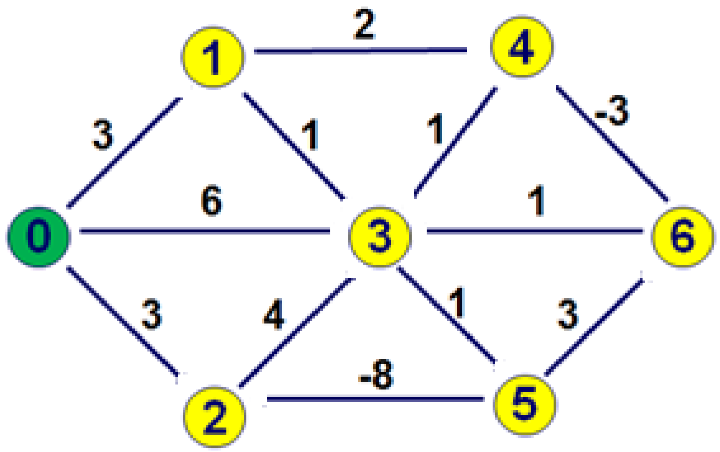

Figure 4.6 An Example of 2-Cycle Elimination

Let l_i^q denote a label associated with node i's state (i, q). Initialize l_i^q and $l_i^q{}'$ for i = 0, 1, 2, 3, 4, 5, 6 and q = 0, 1, 2, 3, 4, 5 according to the step 1 of 2-cycle elimination. Table 4.1 shows the results from the above example.

Table 4.1 Calculations of 2-Cycle Elimination Algorithm

Iteration	Γ	Selected Label	New Generated Labels
1	$l_0^0\left(Null, 0, 0, Strong\right)$	l_0^0	$l_1^1\left(l_0^0, 1, 3, Strong\right)$ $l_2^1\left(l_0^0, 1, 3, Strong\right)$ $l_3^1\left(l_0^0, 1, 6, Strong\right)$
2	$l_1^1\left(l_0^0, 1, 3, Strong\right)$ $l_2^1\left(l_0^0, 1, 3, Strong\right)$ $l_3^1\left(l_0^0, 1, 6, Strong\right)$	$l_1^1\left(l_0^0, 1, 3, Strong\right)$	$l_3^2\left(l_1^1, 2, 4, Strong\right)$ $l_4^2\left(l_1^1, 2, 5, Strong\right)$
		$l_2^1\left(l_0^0, 1, 3, Strong\right)$	$l_3^2{}'\left(l_2^1, 2, 7, Weak\right)$ $l_5^2\left(l_2^1, 2, -5, Strong\right)$

Table 4.1 (continued)

		$l_3^1\left(l_0^0,1,6,Strong\right)$	$l_1^2\left(l_3^1,2,7,Strong\right)$ $l_2^2\left(l_3^1,2,10,Strong\right)$ $l_4^2{}'\left(l_3^1,2,7,Weak\right)$ $l_5^2{}'\left(l_3^1,2,7,Weak\right)$ $l_6^2\left(l_3^1,2,7,Strong\right)$
3	$l_3^2\left(l_1^1,2,4,Strong\right)$ $l_4^2\left(l_1^1,2,5,Strong\right)$ $l_5^2\left(l_2^1,2,-5,Strong\right)$ $l_1^2\left(l_3^1,2,7,Strong\right)$ $l_2^2\left(l_3^1,2,10,Strong\right)$ $l_6^2\left(l_3^1,2,7,Strong\right)$	$l_3^2\left(l_1^1,2,4,Strong\right)$	$l_1^3\left(l_3^2{}',3,8,Strong\right)$ $l_2^3{}'\left(l_3^2,3,8,Weak\right)$ $l_4^3{}'\left(l_3^2,3,5,Weak\right)$ $l_5^3{}'\left(l_3^2,3,5,Weak\right)$ $l_6^3\left(l_3^2,3,5,Dominated\right)$
		$l_4^2\left(l_1^1,2,5,Strong\right)$	$l_1^3{}'\left(l_4^2{}',3,9,Weak\right)$ $l_3^3{}'\left(l_4^2,3,6,Weak\right)$ $l_6^3{}'\left(l_4^2,3,2,Weak\right)$
		$l_5^2\left(l_2^1,2,-5,Strong\right)$	$l_2^3\left(l_5^2{}',3,-1,Strong\right)$ $l_3^3\left(l_5^2,3,-4,Strong\right)$ $l_6^3\left(l_5^2,3,-2,Strong\right)$
		$l_1^2\left(l_3^1,2,7,Strong\right)$	$l_3^3\left(l_1^2{}',3,+\infty,Dominated\right)$ $l_4^3\left(l_1^2,3,9,Dominated\right)$
		$l_2^2\left(l_3^1,2,10,Strong\right)$	$l_3^3\left(l_2^2{}',3,+\infty,Dominated\right)$ $l_5^3\left(l_2^2,3,2,Strong\right)$
		$l_6^2\left(l_3^1,2,7,Strong\right)$	$l_3^3\left(l_6^2{}',3,+\infty,Dominated\right)$ $l_4^3\left(l_6^2,3,4,Strong\right)$ $l_5^3\left(l_6^2,3,10,Dominated\right)$

Table 4.1 (continued)

4	$l_1^3\left(l_3^2{}',3,8,Strong\right)$ $l_2^3\left(l_5^2{}',3,-1,Strong\right)$ $l_3^3\left(l_5^2,3,-4,Strong\right)$ $l_6^3\left(l_5^2,3,-2,Strong\right)$ $l_5^3\left(l_2^2,3,2,Strong\right)$ $l_4^3\left(l_6^2,3,4,Strong\right)$	$l_1^3\left(l_3^2{}',3,8,Strong\right)$	$l_3^4\left(l_1^3{}',4,10,Dominated\right)$ $l_4^4\left(l_1^3,4,10,Dominated\right)$
		$l_2^3\left(l_5^2{}',3,-1,Strong\right)$	$l_3^4{}'\left(l_2^3,4,3,Weak\right)$ $l_5^4\left(l_2^3{}',4,0,Strong\right)$
		$l_3^3\left(l_5^2,3,-4,Strong\right)$	$l_1^4\left(l_3^3,4,-3,Strong\right)*$ $l_2^4{}'\left(l_3^3,4,0,Weak\right)$ $l_4^4{}'\left(l_3^3,4,-3,Weak\right)$ $l_5^4\left(l_3^3{}',4,7,Dominated\right)$ $l_6^4\left(l_3^3,4,-3,Strong\right)$
		$l_6^3\left(l_5^2,3,-2,Strong\right)$	$l_3^4\left(l_6^3,4,-1,Strong\right)$ $l_4^4\left(l_6^3,4,-5,Strong\right)$ $l_5^4{}'\left(l_6^3{}',4,5,Weak\right)$
		$l_5^3\left(l_2^2,3,2,Strong\right)$	$l_2^4\left(l_5^3{}',4,-3,Strong\right)*$ $l_3^4\left(l_5^3,4,3,Dominated\right)$ $l_6^4\left(l_5^3,4,5,Dominated\right)$
		$l_4^3\left(l_6^2,3,4,Strong\right)$	$l_1^4{}'\left(l_4^3,4,6,Weak\right)$ $l_3^4\left(l_4^3,4,5,Dominated\right)$ $l_6^4{}'\left(l_4^3{}',4,2,Weak\right)$

Table 4.1 (continued)

5	$l_5^4\left(l_2^{3\prime},4,0,Strong\right)$ $l_1^4\left(l_3^3,4,-3,Strong\right)$ $l_6^4\left(l_3^3,4,-3,Strong\right)$ $l_3^4\left(l_6^3,4,-1,Strong\right)$ $l_4^4\left(l_6^3,4,-5,Strong\right)$ $l_2^4\left(l_5^{3\prime},4,-3,Strong\right)$	$l_5^4\left(l_2^{3\prime},4,0,Strong\right)$	$l_2^5\left(l_5^{4\prime},5,-3,Strong\right)$ $l_3^{5\prime}\left(l_5^4,5,1,Weak\right)$ $l_6^{5\prime}\left(l_5^4,5,3,Weak\right)$
		$l_1^4\left(l_3^3,4,-3,Strong\right)$	$l_3^5\left(l_1^{4\prime},5,7,Dominated\right)$ $l_4^{5\prime}\left(l_1^4,5,-1,Weak\right)$
		$l_6^4\left(l_3^3,4,-3,Strong\right)$	$l_3^5\left(l_6^{4\prime},5,3,Dominated\right)$ $l_4^5\left(l_6^4,5,-6,Strong\right)*$ $l_5^{5\prime}\left(l_6^4,5,0,Weak\right)$
		$l_3^4\left(l_6^3,4,-1,Strong\right)$	$l_1^{5\prime}\left(l_3^4,5,0,Weak\right)$ $l_2^{5\prime}\left(l_3^4,5,3,Weak\right)$ $l_4^5\left(l_3^4,5,0,Dominated\right)$ $l_5^5\left(l_3^4,5,0,Dominated\right)$ $l_6^5\left(l_3^{4\prime},5,4,Dominated\right)$
		$l_4^4\left(l_6^3,4,-5,Strong\right)$	$l_1^5\left(l_4^4,5,-3,Strong\right)$ $l_3^5\left(l_4^4,5,-4,Strong\right)*$ $l_6^5\left(l_4^{4\prime},5,-6,Strong\right)*$
		$l_2^4\left(l_5^{3\prime},4,-3,Strong\right)$	$l_3^5\left(l_2^4,5,1,Dominated\right)$ $l_5^5\left(l_2^{4\prime},5,-8,Strong\right)*$

Note: * represents the best label for the current state.

In Table 4.1, *Typ* = *Dominated* means that the new generated label is dominated by other labels so that the new generated label will be discarded. The algorithm returns the best label at each node, which represents the shortest path to the associated node. The best labels are marked with an asterisk in Table 4.1. It is clear that 2-cycles are effectively eliminated in the calculation. However, the algorithm cannot eliminate cycles of length > 2. For example, $l_5^5\left(l_2^4{}',5,-8,Strong\right)$ which is the best label at node 5 contains a 3-cycle (5, 3, 2, 5). In order to remove cycles of length > 2, a new resource which indicates the availability of a node is added.

Dummy Resources

Beasley and Christofides (1989) proposed a dummy resource for each node *i*. The dummy resource is binary: each node has one unit of dummy resource and the dummy resource is consumed when the corresponding node is visited. The consumption of the dummy resource effectively prohibits the corresponding node from being visited more than once. In general, *n* dummy resources are required for *n* nodes. The complexity of Beasley and Christofides' approach is $O\left(Qn^2 2^n\right)$. The running time will be exponential in the number of dummy resources, *n*. As a result, this approach was not implemented and no computational experiments were conducted.

In this section, Beasley and Christofides' idea is borrowed by introducing dummy resources for the nodes. However, dummy resources are only implemented to a subset of pickup points. The number of required dummy resources decreased, thus reducing the complexity. This approach combined with 2-cycle elimination algorithm will provide significant bound improvement to the *LRMP*.

The dummy resource is defined as the availability of a node to a path. It is consumed when the corresponding node is visited. Therefore the corresponding node is not available to the extension of a path that consumes the dummy resource. The dummy resource is defined as follows.

$$£_i^r = \begin{cases} 0, \text{ if the corresponding node } i \text{ is visited by path } r, \text{ where } i \in U \\ 1, \text{ otherwise} \end{cases}$$

$£ = (i, r)$ is dummy resource vector and U is a subset of pickup points. Obviously the complexity depends on $|U|$. The subset should include the pickup points that are most possible to form cycles in the path extension procedure. By imposing dummy resources only on those vulnerable pickup points, the algorithm could eliminate cycles as well as reducing computation burden.

In order to determine a proper subset U, solution from previously discussed 2-Cycle elimination algorithm is used. For the routes set R' generated by 2-Cycle elimination, first of all, routes with cycles are identified. Pickup points which constitute cycles in these routes are considered as the most vulnerable nodes that need to be put in U. Then, dummy resources $£$ are imposed to the pickup points in U. Finally, the labels which contain cycles are removed from the solution set.

After modification, the 2-Cycle elimination algorithm will run and solve the problem again. The solution, which may contain cycles, is then examined again and based on the examination, new pickup points that are forming cycles in the solution are added to U. This procedure is repeated until the solution from the 2-Cycle elimination algorithm is elementary. The pseudo code for this combined algorithm is shown as follows.

Step 1. Perform 2-Cycle elimination to ESPPCC. Get routes set R'.

Step 2. If R' is not elementary, go to step 3. Else, stop.

Step 3. Identify nodes forming cycles in R'. Add them to U.

Step 4. Impose dummy resources to nodes in U. Update R'. Go to step 2.

Take the network presented in Figure 4.6 as an example. $l_5^5\left(l_2^4{}',5,-8,Strong\right)$ contains a 3-cycle (0, 2, 5, 3, 2, 5). Node 2 and node 5 are identified as vulnerable nodes that are forming cycles. Therefore, dummy resources are added to node 2 and node 5 and $l_5^5\left(l_2^4{}',5,-8,Strong\right)$ is removed from the solution. After re-run the 2-Cycle algorithm, $l_5^5\left(l_2^4{}',5,-8,Strong\right)$ which corresponds to route (0, 2, 5) is selected as the best label at node 5.

Summary

This section described a column generation based algorithm to solve the CDVRPPD. A labeling algorithm combined with cycle elimination procedure is used to solve the pricing sub-problem. Figure 4.7 shows the overall structure of the algorithm.

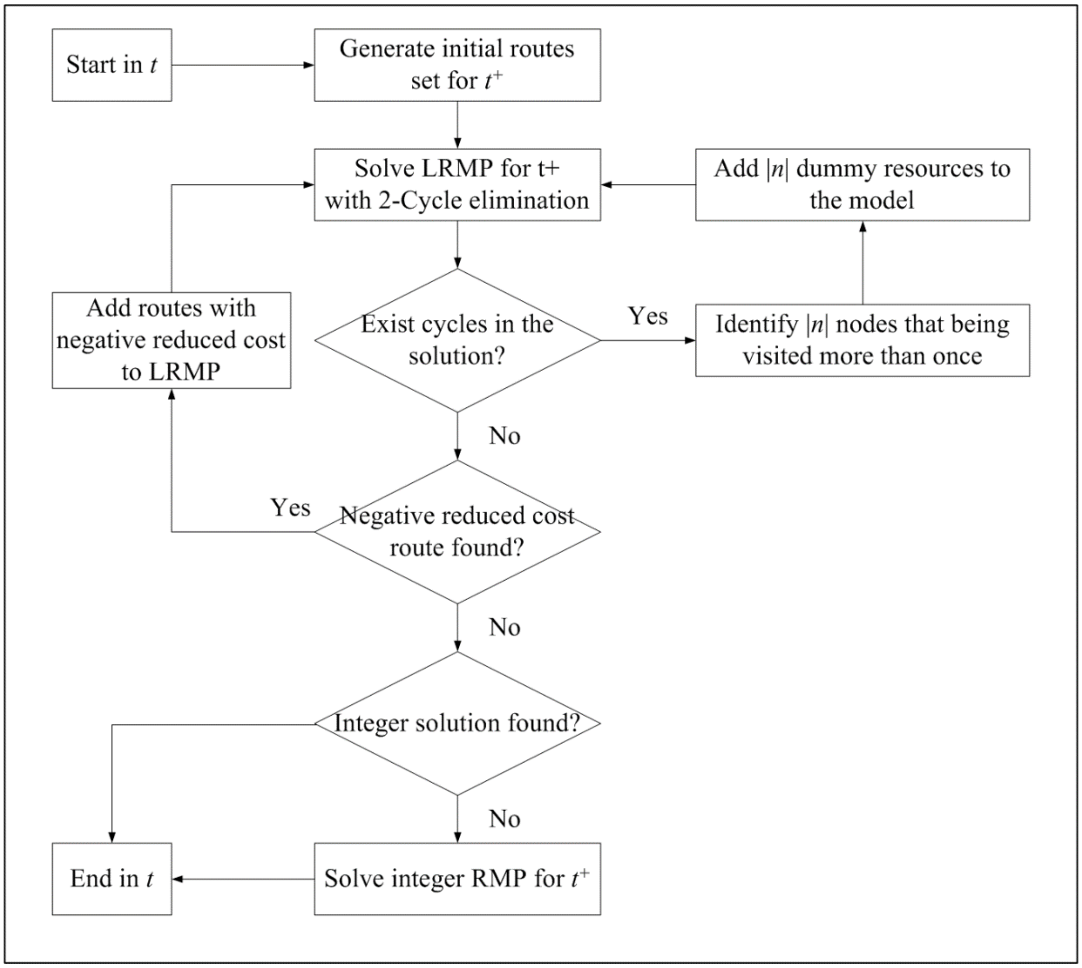

Figure 4.7 Overall Structure of Column Generation Algorithm

In Figure 4.7, one of the key steps is to generate initial routes set for the current interval. This procedure is completed by inserting and deleting evacuee requests from the last interval. Generally, the routes set is expanded constantly along with the increase of customer requests. For example, for a problem with 100 customers, the initial routes set could include more than 20000 routes in the 7th interval. In order to keep the routes set to a manageable size for the following integer problem, the algorithm will limit the size of initial routes set to 10000, which means the initialization process will be stopped as soon as it reaches the limit. The procedure will not affect the performance of the algorithm but save computation time for solving the integer problem.

Computational Results

Benchmark instances available at http://goo.gl/9tclrK were used to evaluate the proposed algorithms. The name of an instance indicates the type of the problem (A, E and S represents Asymmetric, Euclidean and Symmetric problems, respectively) and the number of nodes and available vehicles. The last character denotes the source of the instance. For example, the instance E051-05E is a Euclidean problem proposed by Christofides and Eilon (1969) which consists of 51 nodes and 5 vehicles. In the instances, nodes are distributed in a Euclidean plane. The travel times among nodes are calculated according to the corresponding Euclidean distances.

The instances were performed on an Intel P8200 Duo 2.2 GHz PC with 4G memory. CPLEX was used as the LP and MIP solver. For each instance, the lower bound, the number of columns in R_t, and the total computational time taken in CPU seconds were reported. The results from Agarwal, Mathur, and Salkin (1989), Bixby (1998), and Hadjiconstantinou, Christofides, and Mingozzi (1995) were also presented in Table 4.3 – Table 4.5 in comparison.

Table 4.2 Computational Results

No.	Instance	Nodes	Z^*	Z^{LB}	Effectiveness of Z^{LB}	Cols	Time for Generating Columns (s)	Total CPU Time (s)
1	E016-03m	15	273	270	98.9%	264	1.4	3.9
2	E021-04m	20	353	353	100.0%	492	1.1	3.5
3	E026-08m	25	607	606	99.8%	642	1.0	2.4
4	E031-09h	30	610	605	99.2%	1137	7.5	19.1
5	E036-11h	35	698	698	100.0%	1644	6.5	13.7
6	E041-14H	40	859	859	100.0%	1829	21.0	59.5
7	E051-05e	50	521	518	99.4%	4904	53.1	138.8
8	E076-10e	75	830	815	98.2%	8919	126.5	335.1
9	E101-08e	100	815	804	98.6%	10248	744.8	2381.2
10	E101-10c	100	820	803	97.9%	14346	801.2	2503.2

Table 4.3 Results Comparison of Agarwal, Mathur, and Salkin (AMS) and CE

Problem	n	Z^*	AMS		CE Algorithm		
			Z^{LB}	Effectiveness of Z^{LB}	Z^{LB}	Effectiveness of Z^{LB}	Total CPU Sec
E016-03M	16	273	268	98.2%	270	98.9%	3.9
E021-04M	21	353	351	99.4%	353	100.0%	3.5
E022-04G	22	375	374	99.7%	369	98.4%	1.1
E026-08M	26	607	606	99.8%	606	99.8%	2.4

Table 4.4 Results Comparison of Bixby and CE

Problem	n	Z^*	Bixby		CE Algorithm		
			Z^{LB}	Effectiveness of Z^{LB}	Z^{LB}	Effectiveness of Z^{LB}	Total CPU Sec
E023-03G	23	568	566	99.6%	567	99.8%	23.5
E030-04S	30	503	503	100.0%	503	100.0%	9.5

Table 4.5 Results Comparison of Hadjiconstantinou, Christofides, and Mingozzi (HCM) and CE

Problem	n	Z^*	HCM		CE Algorithm		
			Z^{LB}	Effectiveness of Z^{LB}	Z^{LB}	Effectiveness of Z^{LB}	Total CPU Sec
E036-llH	36	698	694	99.4%	698	100.0%	13.7
E041-14H	41	859	852	99.2%	859	100.0%	59.5
E051-05E	51	521	516	99.0%	518	99.4%	138.8
E076-10E	76	830	815	98.2%	815	98.2%	335.1
E101-08E	101	815	792	97.2%	804	98.7%	2381.2

In Table 4.3 – Table 4.5, the name of the instance and the number of nodes involved are listed. The value of the optimal integer solution Z^*, lower bound Z^{LB}, and the effectiveness of Z^{LB}, which is the difference between the optimum and lower bound, are provided for the AMS, Bixby, HCM methods, and CE algorithm, respectively. In addition, the computational time of the CE algorithm is provided in the table, however, no data is found regarding the computational time of the AMS, Bixby, HCM methods. Among 9 of the 11 instances, the lower bound Z^{LB} provided by the CE algorithm is tighter than the other three methods.

SmartEvac System

To implement the proposed models and algorithms, a SmartEvac system is designed for emergency evacuation. The framework of the SmartEvac system is shown in Figure 4.8. The center part of Figure 4.8 is the optimization module where the models and algorithms are implemented. The CORSIM simulation module on the right hand side provides simulation data to the optimization module. The real world application module shows the perspective of implementation of the proposed models and algorithms in the real world.

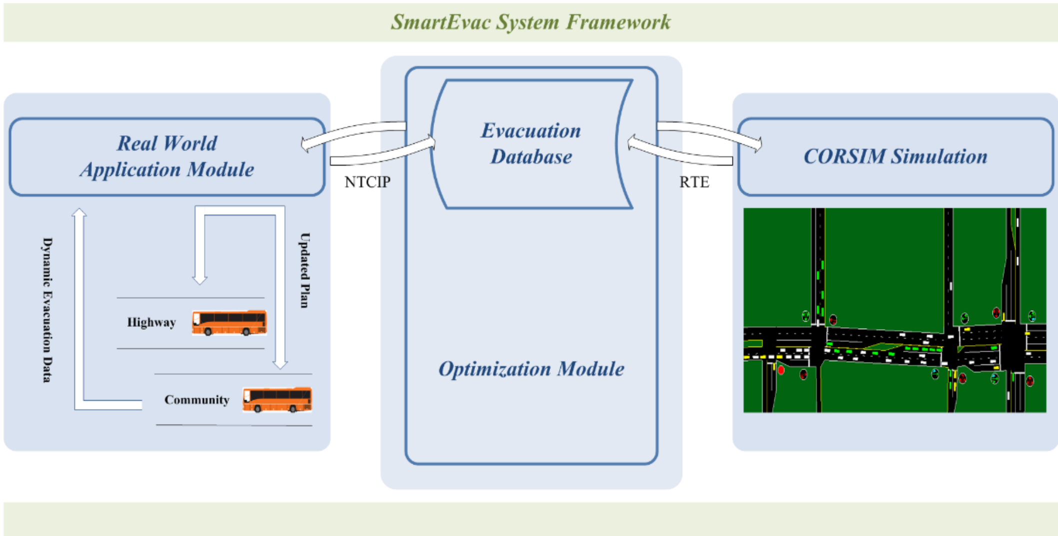

Figure 4.8 SmartEvac System Framework

CORSIM RTE is developed to establish communications between the CORSIM simulation module and the optimization module. At each interval, the CORSIM RTE collects data from the CORSIM simulation, and exports the simulation data to the evacuation database in the optimization module. Furthermore, the CORSIM RTE is programmed to monitor network interruptions that may have significant impacts on travel time, and capture the drastic travel time change. In addition to the simulation data, the evacuation database is capable of receiving real-time pickup requests from external data sources. The optimization module utilizes the simulation data, including link travel times and vehicle positions, and the evacuation data, including registered and unregistered evacuee's requests, shelters information, and transit vehicles information, to update the transit vehicle routing plan, and then feeds the updated transit vehicle routing plan back to the simulation module through CORSIM RTE. This process repeats until the emergency evacuation (simulation) is finished. At the end of the evacuation, the

SmartEvac system will create a performance evaluation report in terms of the quality and efficiency of the transit vehicle routing plan.

The SmartEvac system is programmed using Microsoft Visual Studio C++. Microsoft SQL server 2005 is used to accommodate the evacuation database. The class view and functions of the optimization module are shown in Figure 4.9. There are five major classes developed in the system. The CPLEX class is the master class which the optimization models and algorithms are implemented. The CEvacuation class controls the communication between the system and simulation. CORSIM RTE is used as the interface between the system and simulation. CNode, CRoute, and CVehicle classes contain the information of nodes, routes, and vehicles, respectively.

Class: CPLEX	
Initialization ()	: Initialize system parameters
InitializeRoutes ()	: Initialize routes set
MasterModel ()	: Build master problem model
SubModel ()	: Build sub-problem model
LabelInitialization()	: Initialize labels
LabelExtension ()	: Generate labels for each node
DominanceCheck ()	: Apply dominance rule to the labels
IdentifyCycle ()	: Find nodes which form a cycle
AddResource ()	: Add new resources
Integral ()	: Add integer constraints to the model
UpdateNetwork ()	: Update the graph
AddNode ()	: Add a new node in the network
DropNode ()	: Drop a node in the network
PredictTravelTime ()	: Predict travel time for the next interval
UpdateRoutes ()	: Update the vehicles routes
UpdateModel ()	: Update sub-problem model

Class: CEvacuation	
NewPickupRequest ()	: Collect new pickup request
DroppedPickupRequest ()	: Collect dropped pickup request
UpdateTravelTime ()	: Collect dynamic travel time
UpdateVehclePosition ()	: Collect current vehicle's postion

Class: CNode	
GetID ()	: Return node ID
GetType ()	: Return node type
GetDemand ()	: Return node demand
GetStatus ()	: Return node status

Class: CRoute	
GetID ()	: Return route ID
GetCost ()	: Return route cost
GetLoad ()	: Return route load
GetVehicle ()	: Return associated vehicle
GetNodeList ()	: Return node list

Class: CVehicle	
GetID ()	: Return vehicle ID
GetPosition ()	: Return vehicle position
GetLoad ()	: Return vehicle load

Figure 4.9 Class View and Functions of the Optimization Module

The data dictionary of the evacuation database is shown in Figure 4.10. The evacuation database is updated in real time through CORSIM RTE. There are six tables which constitute the database. All the evacuation data regarding the vehicles, shelters, pickup points, evacuees, routes, and travel times are included in the database.

Table: Vehicle_Info	
Vehicle_Id	: Vehicle's ID
Vehicle_Type	: Vehicle's type
Vehicle_Location	: Vehicle's location
Vehicle_Capacity	: Vehicle's capacity
Vehicle_Driver	: Assigned driver to this vehicle

Table: Shelter_Info	
Shelter_Id	: Shelter's ID
Shelter_Name	: Shelter's name
Shelter_Address	: Shelter's address
Shelter_Capacity	: Shelter's capacity

Table: PickupPoint_Info	
PickupPoint_Id	: PickupPoint's ID
PickupPoint_Type	: PickupPoint's type
PickupPoint_Address	: PickupPoint's address
PickupPoint_Demand	: PickupPoint's demand

Table: Evacuee_Info	
Evacuee_Id	: Evacuee's ID
Evacuee_Type	: Evacuee's Type
Evacuee_Name	: Evacuee's name
Evacuee_Address	: Evacuee's address
Evacuee_Phone	: Evacuee's cell phone number
Evacuee_SpecialNeeds	: Evacuee's special requirement

Table: TravelTime_Info	
TravelTime_UPS_Node	: Link upstream node
TravelTime_DNS_Node	: Link downstream node
TravelTime_Time	: Link travel time
TravelTime_Interval	: Interval number

Table: Route_Info	
Route_Id	: Route's ID
Route_Assigned_Vehicle	: Assigned vehicle
Route_Assigned_Node	: Assigned nodes
Route_DepartureTime	: Departure time
Route_ArrivingTime	: Arrival Time

Figure 4.10 Database Table Structure

CHAPTER V

CASE STUDY

In this chapter, a case study of Hurricane Gustavo evacuation in Gulfport is proposed to evaluate the SmartEvac system. The case study is based on CORSIM simulation, which provides dynamic travel time for the system. Scenarios corresponding to different evacuation situations are built in the simulation. The capability of the SmartEvac system working in a dynamic environment is validated by the case study.

Problem Statement

In this case study, emergency evacuation scenarios are replicated based on the data from the Hurricane Gustavo emergency evacuation in 2008. There are 182 registered evacuees across 66 pickup points in the Mississippi Gulf Coast region. In addition, based on CTA's experience, 46 unregistered evacuees across 30 pickup points are considered in this case study. The unregistered evacuees are expected to call for help at any time during the emergency evacuation. Three shelters in the region provide temporary housing for the evacuees. The distribution of shelters and pickup points with registered evacuees is shown in Figure 5.1.

Figure 5.1 Distribution of Shelters and Pickup Points with Registered Evacuees

Detail information about shelters and pickup points with registered evacuees is listed in Table 5.1. Node 0 is the depot where all the transit vehicles depart. Node 1 to Node 3 are the shelters to accommodate the evacuees. Each shelter has a capacity of 350. Node 4 to Node 69 are the pickup points with registered evacuees.

Table 5.1 Shelters and Pickup Points' Information

No.	Address	Demand / Capacity	No.	Address	Demand / Capacity
0	DeBuys Road, Gulfport	NA	35	Hewes Ave., Gulfport	1
1	Auto Mall Pkwy, D'Iberville	350	36	28th St., Gulfport	5
2	Espy Ave., Pass Christian	350	37	Pass Rd., Gulfport	1
3	Eisenhower Dr, Biloxi	350	38	61st Ave., Gulfport	1
4	Bradford St., Biloxi	1	39	Mill Rd., Gulfport	2
5	McDonnell Ave., Biloxi	4	40	Railroad St., Gulfport	2
6	Auburn Dr., Biloxi	1	41	Taylor Blvd., Gulfport	3
7	Maple St., Biloxi	5	42	28th St., Gulfport	4
8	Claiborne St., Biloxi	1	43	32nd Ave., Gulfport	1
9	Hope St., Biloxi	3	44	W Pine St., Gulfport	1
10	Lawrence St., Biloxi	1	45	46th Ave., Gulfport	10
11	Division St., Biloxi	5	46	14th Ave., Gulfport	1
12	Pringle Circle, Biloxi	3	47	Pine Ave., Gulfport	1
13	Hiller Dr., Biloxi	2	48	7th Ave., Gulfport	12
14	Claiborne St., Biloxi	2	49	South Carolina Ave., Gulfport	2
15	Nichols Dr., Biloxi	3	50	Cuandet Rd., Gulfport	1
16	Benachi Ave., Biloxi	5	51	53rd Ave., Gulfport	2
17	Popps Ferry Rd., Biloxi	1	52	19th St., Gulfport	1
18	Water St., Biloxi	1	53	28th St., Gulfport	1
19	Roy St., Biloxi	2	54	Railroad St., Gulfport	2
20	Atkinson Rd., Biloxi	2	55	39TH Ave., Gulfport	2
21	Auburn Dr., Biloxi	2	56	Fournier Ave., Gulfport	2
22	Strangi Ave., Biloxi	2	57	Halsell Rd., Gulfport	1
23	Pass Rd., Biloxi	2	58	Pass Rd., Gulfport	2
24	Atkinson Rd., Biloxi	14	59	Tegarden Rd, Gulfport	4
25	Pear Dr., Biloxi	1	60	East Augustus St., D'Iberville	1
26	Judge Sekul Ave., Biloxi	15	61	Popps Ferry Rd., D'Iberville	2
27	Beach Blvd., Biloxi	1	62	Cedar Dr., D'Iberville	6
28	Oneal Rd., Gulfport	1	63	Cedar Dr., D'Iberville	1
29	Tara Hills Dr., Gulfport	1	64	Railroad St., Long Beach	4
30	19th St., Gulfport	3	65	28th St., Long Beach	1
31	Central Ave., Gulfport	1	66	Middle Ave Ocean Springs	3
32	34th St., Gulfport	3	67	Popps Ferry Rd., Biloxi	2
33	Ohio Ave., Gulfport	2	68	Pass Rd., Gulfport	3
34	8010 Hwy 49, Gulfport	1	69	Dedeaux Rd., Gulfport	3

A homogenous fleet of transit vehicle is used in the emergency evacuation. The capacity of the transit vehicle is 30. Each transit vehicle has an onboard equipment that is able to receive orders from the SmartEvac system in real time. The dwell time at each pickup point is two minutes.

According to the CTA's evacuation plan, the emergency evacuation started at 7:00 AM rush hour. It is assumed that calls from the unregistered evacuees will evenly arrive with 3-minute interval.

CORSIM Simulation Development

The transportation network data and evacuation data used to build the CORSIM network are collected from field survey, CTA, and the Office of Engineering, etc. Figure 5.2 shows the data used in the simulation.

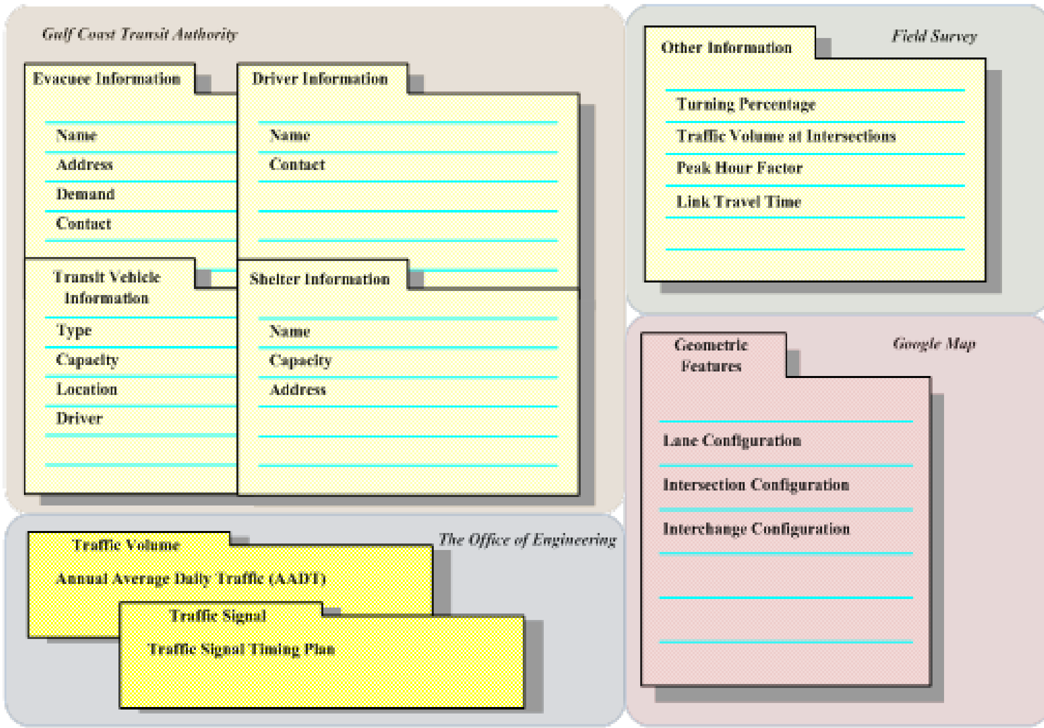

Figure 5.2 Data Used in the CORSIM Simulation

Field surveys were conducted at 23 major intersections in the Gulfport Coast region. These intersections are mainly distributed along Pass Road, Highway 605, Canal Road, and Popps Ferry Road. Radar detectors and manual counters were deployed at the 23 intersections for five days to collect daily traffic volumes, peak hour traffic volumes, and turning percentage data. Turn prohibitions are implemented at specific intersections where prohibitory traffic signs are placed.

The CORSIM network is shown in Figure 5.3. Interstate 10 runs east and west of the Mississippi Gulf Coast region. Other major roadways include I-110, U.S.90, U.S.49, Pass Road, Highway 605, and Highway 67. The CORSIM network consists of 1,632 links and 1,341 nodes, in which 146 nodes are signalized intersections. The traffic signal timing plans were extracted from the City Engineering ACTRA system.

Figure 5.3 CORSIM Network of Gulfport Region

In addition to the intersections and transition nodes, depots, shelters, and pickup points are coded in the CORSIM network. In Figure 5.3, depots, shelters, and pickup points with registered evacuees are marked with yellow, red, and blue color in the CORSIM network, respectively. The shortest travel times among vertices including depots, shelters, and pickup points are calculated using a modified Dijkstra Algorithm (Wen, 2012) in which turn prohibitions are considered.

The length of the simulation is two hours which is consistent with the CTA's evacuation plan. Thirty intervals, $t_1, t_2, \cdots, t_{30}$, with equal length of three minutes are implemented in the simulation.

The CORSIM simulation model is fine-tuned with morning rush hour travel time data collected in the field. The simulated travel times on two major roads in the area, U.S. 90 segment between the Bay St. Louis Bridge and the Biloxi Bay Bridge, and Pass Road segment between U.S. 49 and Rodeo Drive, compared with those from Google Map and historical data of 2008, are shown in Figure 5.4.

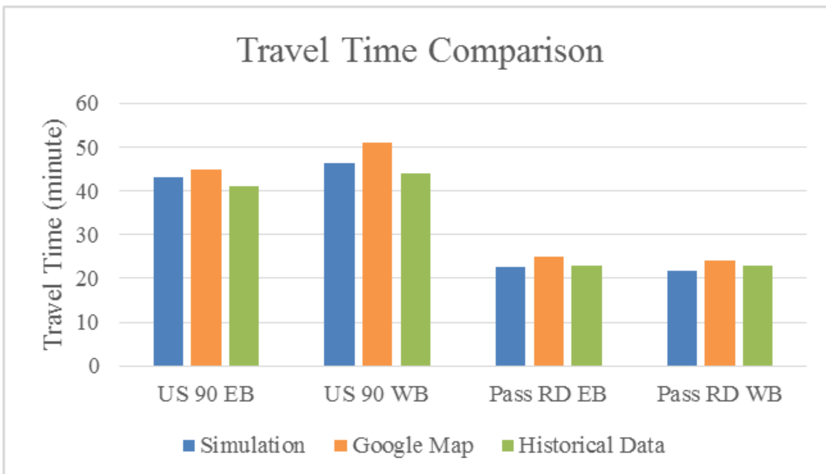

Figure 5.4 Travel Time Comparison between Simulation, Google Map, and Historical Data

Notes: EB – Eastbound, WB – Westbound

Results of Case Study

To evaluate the performance of the SmartEvac system in an emergency evacuation, especially when dynamic factors, such as unregistered evacuees' pickup requests and network interruptions, are considered, the following emergency evacuation scenarios are developed.

Scenario 1

Scenario 1 is developed as a base scenario. There are no dynamic factors in the emergency evacuation which means that the transit vehicle routes remain fixed all the time.

Results from the SmartEvac system are displayed in Figure 5.5. There are seven transit vehicles used in the emergency evacuation. The total travel time of all the seven transit vehicle routes is 417.9 minutes. The total computation time is 157 seconds while the time for generating columns is 68 seconds.

Figure 5.5 Results from the Scenario 1

Each individual transit vehicle route is listed as follows.

Route 1: cost = 40.5 minutes and load = 30.

Node 0 - Node 27 - Node 26 - Node 16 - Node 11 - Node 4 - Node 9 - Node 1

Route 2: cost = 43.0 minutes and load = 16

Node 0 - Node 6 - Node 21 - Node 17 - Node 67 - Node 61 - Node 60 - Node 63 - Node 62 - Node 1

Route 3: cost = 20.4 minutes and load = 18

Node 0 - Node 24 - Node 20 - Node 23 - Node 3

Route 4: cost = 48.4 minutes and load = 29

Node 0 - Node 59 - Node 37 - Node 48 - Node 32 - Node 68 - Node 36 - Node 47 - Node 3

Route 5: cost = 79.9 minutes and load = 30

Node 0 - Node 10 - Node 13 - Node 5 - Node 12 - Node 18 - Node 8 - Node 14 - Node 15 - Node 19 - Node 22 - Node 7 - Node 66 - Node 25 - Node 1

Route 6: cost = 114.1 minutes and load = 29

Node 0 - Node 44 - Node 39 - Node 50 - Node 57 - Node 69 - Node 29 - Node 28 - Node 34 - Node 49 - Node 33 - Node 55 - Node 42 - Node 53 - Node 51 - Node 65 - Node 64 - Node 2

Route 7: cost = 71.7 minutes and load = 30

Node 0 - Node 41 - Node 31 - Node 35 - Node 40 - Node 52 - Node 30 - Node 46 - Node 58 - Node 45 - Node 43 - Node 54 - Node 56 - Node 38 - Node 2

Scenario 2

In order to replicate the scenario that unregistered evacuees call for pickup after the emergency evacuation starts, scenario 2 is created based on scenario 1 but new pickup requests from unregistered evacuees are generated per interval.

Take interval t_1 as an example. In t_1, a new pickup request at node 70 with demand of 1 is added in the system. In response to the new request, the SmartEvac system re-optimizes the transit vehicle routes in t_2. After optimization, three out of the seven routes are adjusted. The updated transit vehicle routes will be implemented in t_3, as shown in Figure 5.6.

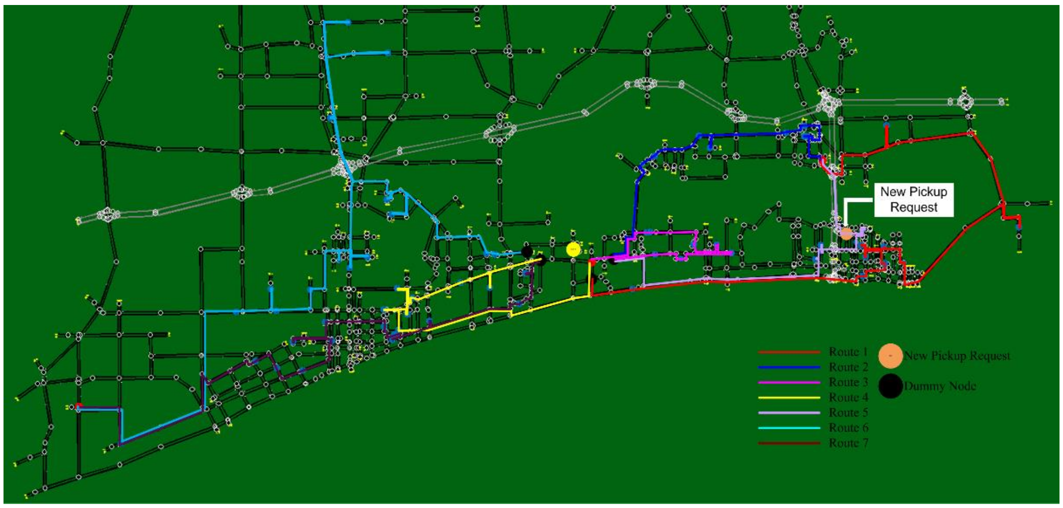

Figure 5.6 Transit Vehicle Routes after Re-optimization in Scenario 2 t_1

The total cost of the re-optimized vehicle routes is 398.7 minutes. There are still seven transit vehicles used in the emergency evacuation. The computation time for the re-optimization is 173 seconds. In comparison with the transit vehicle routes in scenario 1, route 1, route 3, and route 5 are re-optimized due to the new pickup request at node 70. The revisions are shown as follows.

Route 1: cost = 65.6 minutes and load = 21

Dummy Node - Node 27 - Node 18 - Node 8 - Node 14 - Node 15 - Node 19 - Node 22 - Node 7 - Node 66 - Node 25 - Node 1

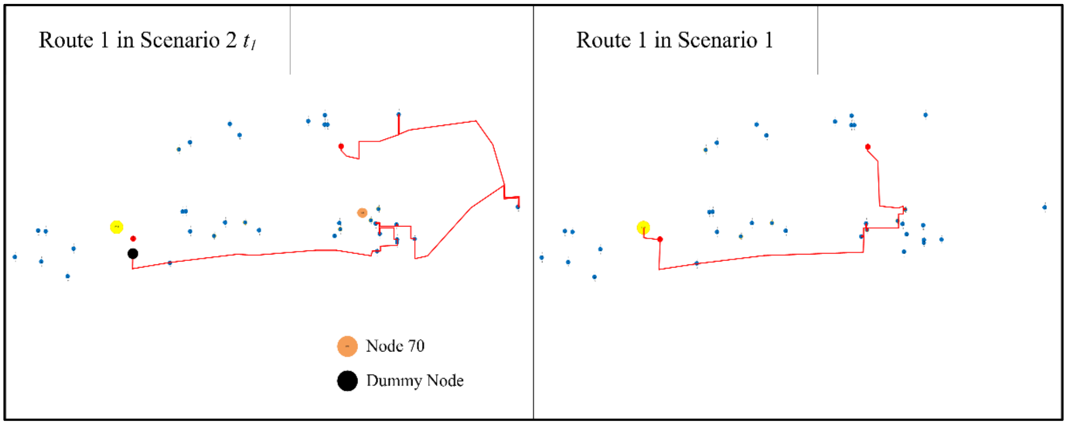

Figure 5.7 Comparison of Route 1 between Scenario 1 and Scenario 2 t_1

Route 3: cost = 33.4 minutes and load = 28

Dummy Node - Node 23 - Node 10 - Node 12 - Node 5 - Node 13 - Node 20 - Node 24 - Node 3

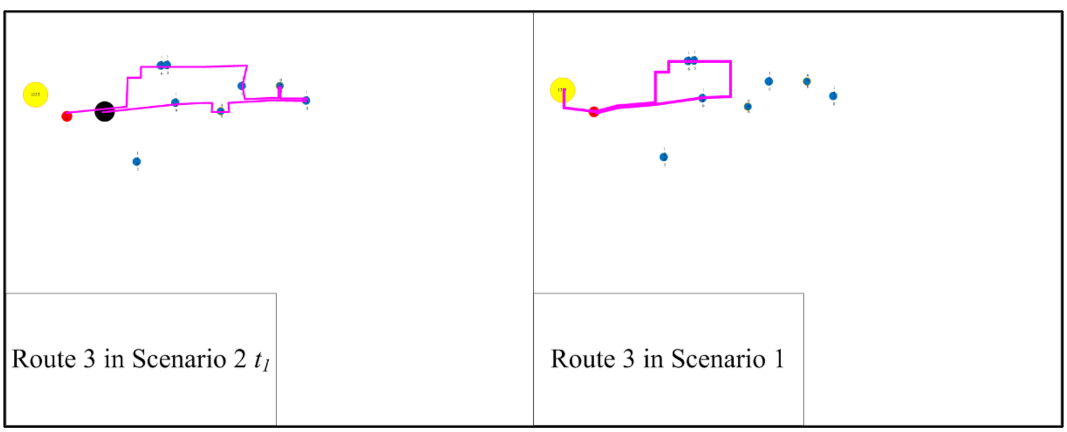

Figure 5.8 Comparison of Route 3 between Scenario 1 and Scenario 2 t_1

Route 5: cost = 34.5 minutes and load = 30

Dummy Node - Node 26 - Node 16 - Node 11 - Node 4 - Node 9 - Node 70 - Node 1

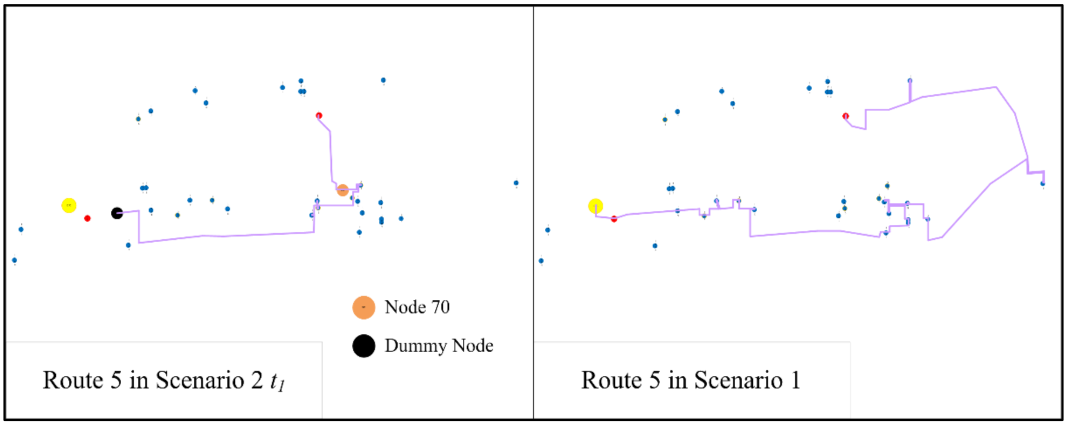

Figure 5.9 Comparison of Route 5 between Scenario 1 and Scenario 2 t_1

The new added pickup request at node 70 is serviced by route 5. In order to explicitly explain how the SmartEvac system adjusts the vehicle routes for the new pickup request, all the pickup points are distributed to eight zones based on their geographic location as shown in Figure 5.10. The boundaries of the zones consist of major roads and bridges in the region, such as I-110, U.S.90, U.S.49, Pass Road, Popps Ferry Bridge.

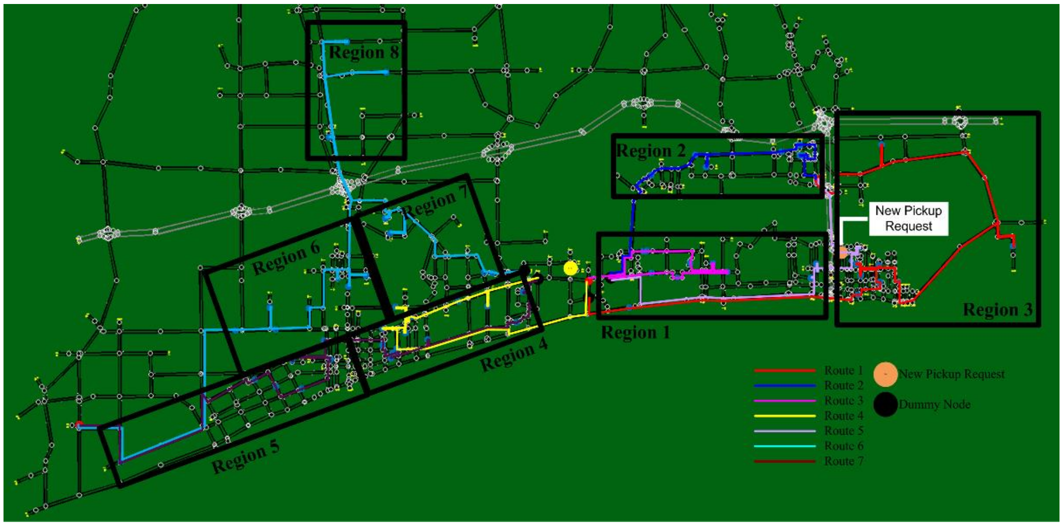

Figure 5.10 The Spatial Distribution of Zones of Pickup Points

The new pickup request, node 70, is located in zone 3. All the pickup points in zone 3 are serviced by two vehicles in scenario 1, vehicle 1 on route 1 and vehicle 5 on route 5. Obviously, vehicle 1 and vehicle 5 are two candidates to pick up node 70. However, both of vehicle 1 and vehicle 5 are full load and no space left for node 70 according to the original routing plan in scenario 1. Therefore, another vehicle is required to partake of vehicle 1 or vehicle 5's load in order to free up space for node 70. As shown in Figure 5.5, there is only one vehicle route, route 3 in zone 1, which covers route 5. No vehicle routes can cover route 1 by making trivial revisions. Both of route 3 and route 5 include Pass Road section from Popps Ferry Road to Veterans Avenue. The SmartEvac system is able to reassign the pickup points in this section from route 5 to route 3 to release vehicle 5's capacity. Now vehicle 5 has sufficient capacity to pick up node 70 because node 10, node 13, node 5, and node 12 are taken over by vehicle 3.

However, vehicle 5 is still not the best option to pick up node 70 because node 70 is farther away from route 5 than route 1. Hence the SmartEvac system swaps vehicle 5's tasks with vehicle 1's tasks. After swap, vehicle 1's tasks after picking up node 27 are taken over by vehicle 5 and all vehicle 5's tasks are taken over by vehicle 1. Finally, vehicle 5 turns to be the most appropriate vehicle to pick up node 70 and route 5 is revised by including node 70.

A new pickup request per interval is received for 30 intervals. The SmartEvac system updates the pickup information and re-optimizes the transit vehicle routes accordingly. The results are summarized in Table 5.2.

Table 5.2 Results of Scenario 2 with Fixed Interval

Time Interval	Total Cost (Minute)	No. of Vehicle	Computation Time (Second)
t_0	417.9	7	157
t_1	398.7	7	171
t_2	377.3	7	168
t_3	345.3	7	140
t_4	317.4	7	121
t_5	294.6	7	48
t_6	278.0	7	31
t_7	236.4	6	12
t_8	219.6	6	10
t_9	192.5	5	3
t_{10}	177.0	5	2
t_{11}	175.9	5	4
t_{12}	157.0	5	2
t_{13}	154.4	5	2
t_{14}	147.7	5	2
t_{15}	143.7	5	3
t_{16}	130.5	4	6
t_{17}	113.6	3	1
t_{18}	115.2	3	1
t_{19}	100.3	3	1
t_{20}	89.3	3	1
t_{21}	93.3	3	1
t_{22}	98.5	3	1
t_{23}	110.6	3	1
t_{24}	101.8	2	1
t_{25}	92.1	2	1
t_{26}	97.0	2	1
t_{27}	91.8	2	1
t_{28}	93.2	2	1
t_{29}	110.1	3	1
t_{30}	118.7	3	1

As discussed in Chapter III, dynamic intervals could be implemented in the optimization process. The length of interval t_i is calculated based on the computation time in t_{i-1}. The initial time interval is 180 seconds and the minimum time interval is 60 seconds. The incremental factor β is 110%. The results of Scenario 2 with dynamic intervals are listed in Table 5.3. Table 5.3 only shows the intervals in which the SmartEvac processing the new pickup requests.

Table 5.3 Results of Scenario 2 with Dynamic Interval

Total Cost (Minute)	Computation Time (Second)	Interval Length (Second)	Wait Time (Second)	New Request Arrival Time (Second)
417.9	156	180		90
399.7	171	172	262	285
376.3	165	188	255	472
345.2	137	182	249	613
320.8	119	151	259	790
296.5	51	131	213	995
289.9	34	60	68	1155
248.4	15	60	97	1345
231.6	11	60	87	1530
202.5	5	60	82	1680
187.0	5	60	112	1900
185.9	4	60	72	2083
167.0	3	60	69	2257
164.4	2	60	75	2433
157.7	6	60	79	2587
153.7	5	60	105	2801
138.5	5	60	71	2979
119.6	4	60	73	3122
121.2	3	60	110	3350
106.3	1	60	62	3518
95.3	1	60	74	3701
99.3	1	60	71	3891
104.5	1	60	61	4023
116.6	1	60	109	4225
105.8	1	60	87	4419
96.1	1	60	73	4581
101.0	1	60	91	4770
95.8	1	60	82	4931
97.2	1	60	101	5128
116.1	1	60	84	5331
124.7	1	60	61	

In scenario 2, there is a total of 30 requests from unregistered evacuees added in the emergency evacuation. Figure 5.11 shows the distribution of all evacuees, including both registered and unregistered evacuees.

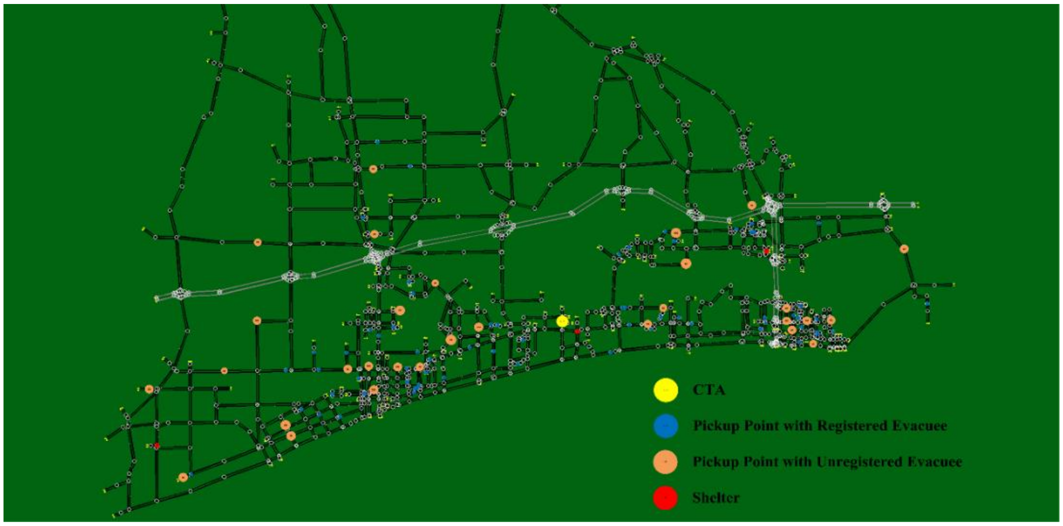

Figure 5.11 CORSIM Network with Unregistered Evacuees

Scenario 3

Scenario 3 is developed base on scenario 2 but certain incidents, such as traffic accidents, bridge broken, are implemented.

Scenario 3(a)

Assuming that traffic accidents occur on U.S. 90 after the emergency evacuation starts, as shown in Figure 5.12, the travel speed on U.S. 90 is severely impacted by the accidents. It's assumed that the average travel time on U.S. 90 in scenario 3 is twice of what's in scenario 2.

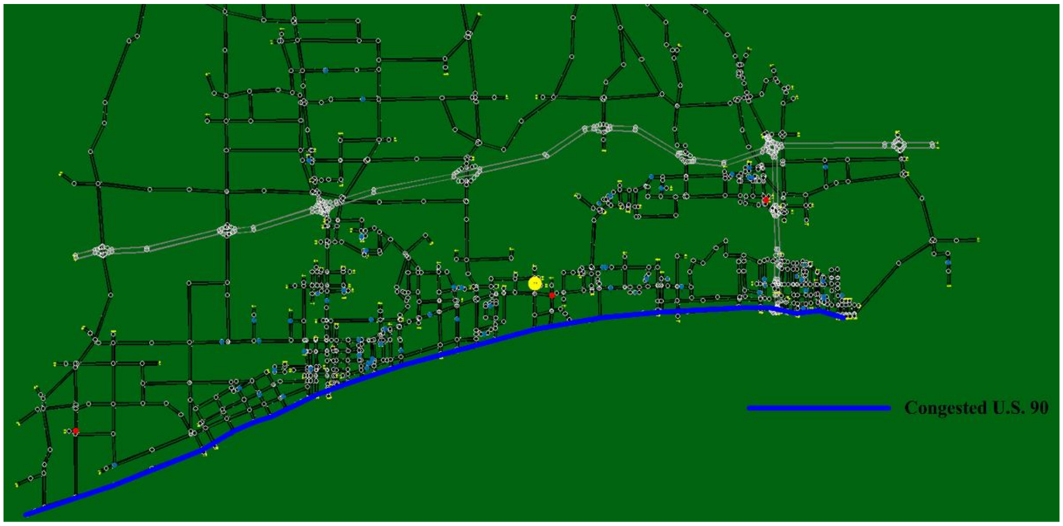

Figure 5.12 CORSIM Network of Scenario 3(a)

The SmartEvac system is able to capture the travel time surge in real time and update the transit vehicle routes accordingly. Assume that the travel time surge happens in t_1, the updated results comparing with the results from scenario 2 are presented as follows.

Results in scenario 3(a) t_1: the total travel time is 408.6 minutes and the computation time is 173 seconds.

Route 1: cost = 74.2 minutes and load = 21

Dummy Node - Node 27 - Node 18 - Node 8 - Node 14 - Node 15 - Node 19 - Node 22 - Node 7 - Node 66 - Node 25 - Node 1

Route 2: cost = 40.0 minutes and load = 16

Dummy Node - Node 6 - Node 21 - Node 17 - Node 67 - Node 61 - Node 60 - Node 63 - Node 62 - Node 1

Route 3: cost = 33.7 minutes and load = 28

Dummy Node - Node 23 - Node 10 - Node 12 - Node 5 - Node 13 - Node 20 - Node 24 - Node 3

Route 4: cost = 44.7 minutes and load = 29

Dummy Node - Node 59 - Node 47 - Node 36 - Node 68 - Node 32 - Node 48 - Node 37 - Node 3

Route 5: cost = 36.8 minutes and load = 30

Dummy Node - Node 26 - Node 16 - Node 11 - Node 4 - Node 9 - Node 70 - Node 1

Route 6: cost = 79.1 minutes and load = 29

Dummy Node - Node 44 - Node 39 - Node 50 - Node 57 - Node 69 - Node 29 - Node 28 - Node 34 - Node 49 - Node 33 - Node 55 - Node 42 - Node 53 - Node 51 - Node 65 - Node 64 - Node 2

Route 7: cost = 42.1 minutes and load = 30

Dummy Node - Node 41 - Node 31 - Node 35 - Node 40 - Node 52 - Node 30 - Node 46 - Node 58 - Node 45 - Node 43 - Node 54 - Node 56 - Node 38 - Node 2

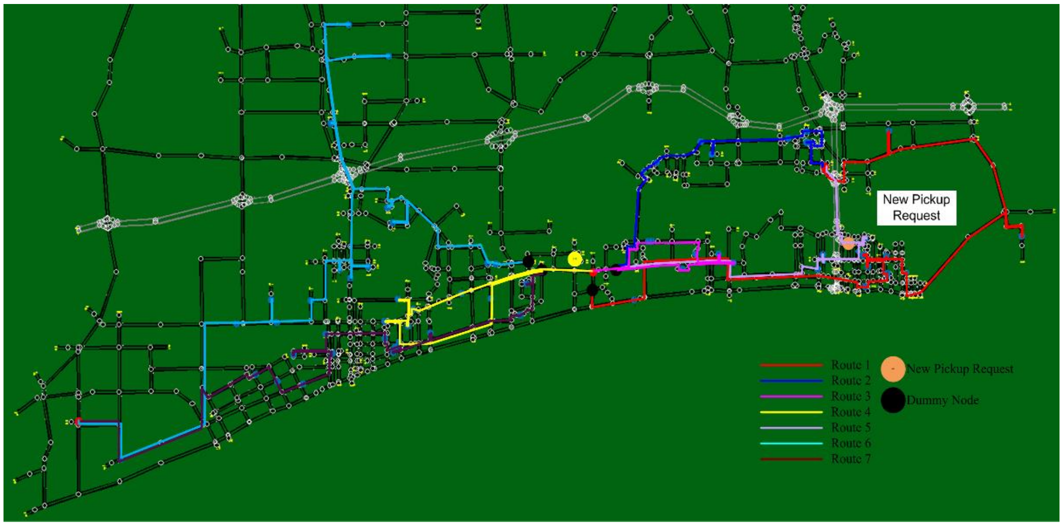

Figure 5.13 Transit Vehicle Routes after Re-optimization in Scenario 3(a) Interval t_1

Three routes, route 1, route 4, and route 5, are revised due to congestions on U.S. 90. See Figure 5.14 – Figure 5.16 for a comparison of the results between scenario 2 and scenario 3(a).

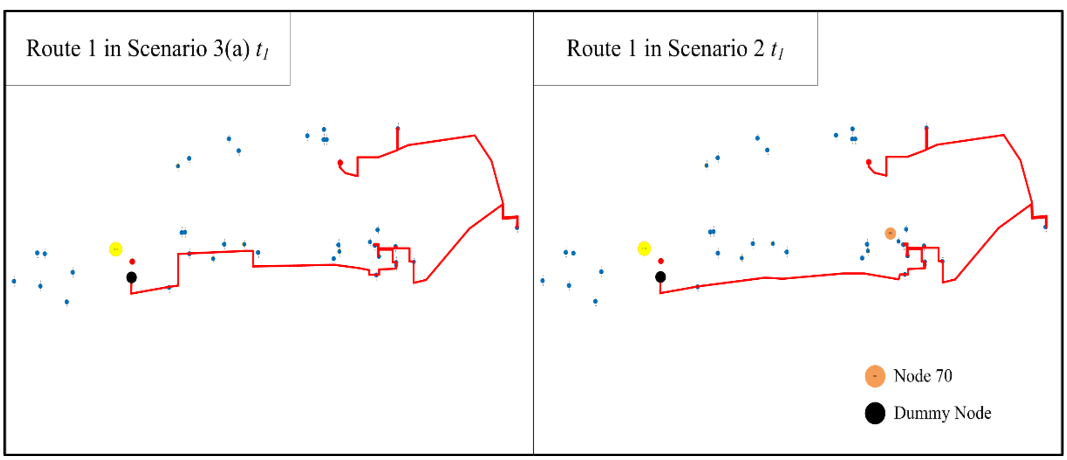

Figure 5.14 Comparison of Route 1 between Scenario 2 t_1 and Scenario 3(a) t_1

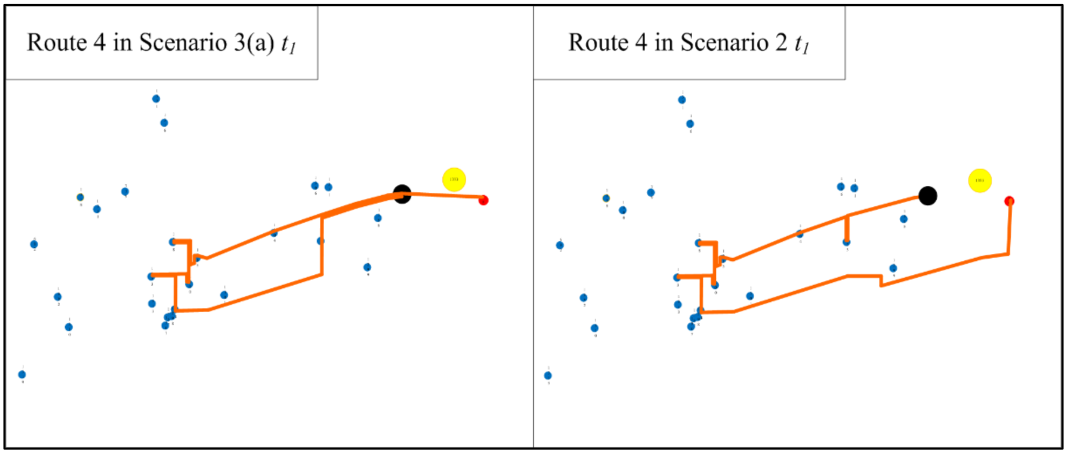

Figure 5.15 Comparison of Route 4 between Scenario 2 t_1 and Scenario 3(a) t_1

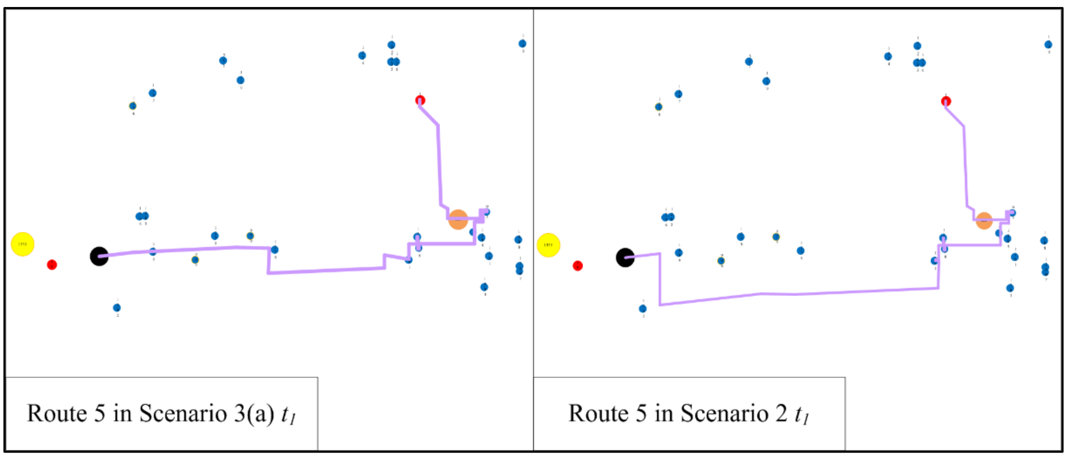

Figure 5.16 Comparison of Route 5 between Scenario 2 t_1 and Scenario 3(a) t_1

In response to the congestions of U.S. 90, the SmartEvac system re-optimize the transit vehicle routes in real time. Since route 1's travel time will increase from 65.6 minutes to 81.7 minutes due to the congestions of U.S. 90, route 1' section of U.S. 90 from Eisenhower Drive to Bellman Street is detoured at Beauvoir Road. Vehicle 1 will be diverted to Pass Road, Irish Hill Drive, and Howard Ave which are parallel to U.S. 90.

The travel time of route 1 decreases from 81.8 minutes to 74.2 minutes through detour. Similarly, route 4's section of U.S. 90 from Tegarden Road to Eisenhower Driver is detoured at Tegarden Road. Vehicle 4 will be diverted to Pass Road. The travel time of route 4 decreases from 50.3 minutes to 44.7 minutes. Route 5's section of U.S. 90 from Beauvoir Road to Porte Avenue is detoured at Beauvoir Road. Vehicle 5 will be diverted to Pass Road and Irish Hill Drive. The travel time of route 5 will reduced from 43.7 minutes and 36.8 minutes. In summary, the total travel time saving from the detour on route 1, route 4, and route 5 is 20.1 minutes.

The rest of results in scenario 3(a) from t_3 to t_{30} are listed in Table 5.4.

Table 5.4 Results of Scenario 3(a)

Time Interval	Total Cost (*Minute*)	No. of Vehicle	Computation Time (*Second*)
t_0	417.9	7	159
t_1	407.3	7	173
t_2	391.6	7	178
t_3	362.7	7	150
t_4	333.7	7	128
t_5	310.2	7	57
t_6	298.9	7	43
t_7	257.1	7	48
t_8	248.2	6	45
t_9	231.7	5	27
t_{10}	207.4	5	13
t_{11}	176.5	5	8
t_{12}	160.4	5	3
t_{13}	157.8	5	1
t_{14}	174.8	5	2
t_{15}	165.1	5	4
t_{16}	158.9	4	2
t_{17}	146.9	4	2
t_{18}	145.8	4	3
t_{19}	134.5	4	3
t_{20}	127.1	4	2
t_{21}	130.5	5	2
t_{22}	122.5	4	1
t_{23}	115.5	4	1
t_{24}	102.5	3	1
t_{25}	95.7	3	1
t_{26}	101.3	2	1
t_{27}	93.2	2	1
t_{28}	94.2	2	1
t_{29}	110.9	3	1
t_{30}	119.6	3	1

Scenario 3(b)

Scenario 3(b) is developed based on scenario 3(a) but in addition to the incidents on U.S. 90, the Biloxi Bay Bridge is assumed to be broken from t_1, which corresponds to the actual situation in Hurricane Gustav. The Biloxi Bay Bridge carries U.S. 90 over

Biloxi Bay between Biloxi and Ocean Springs, as shown in Figure 5.17. Route 1 passes the Biloxi Bay Bridge in scenario 3(a).

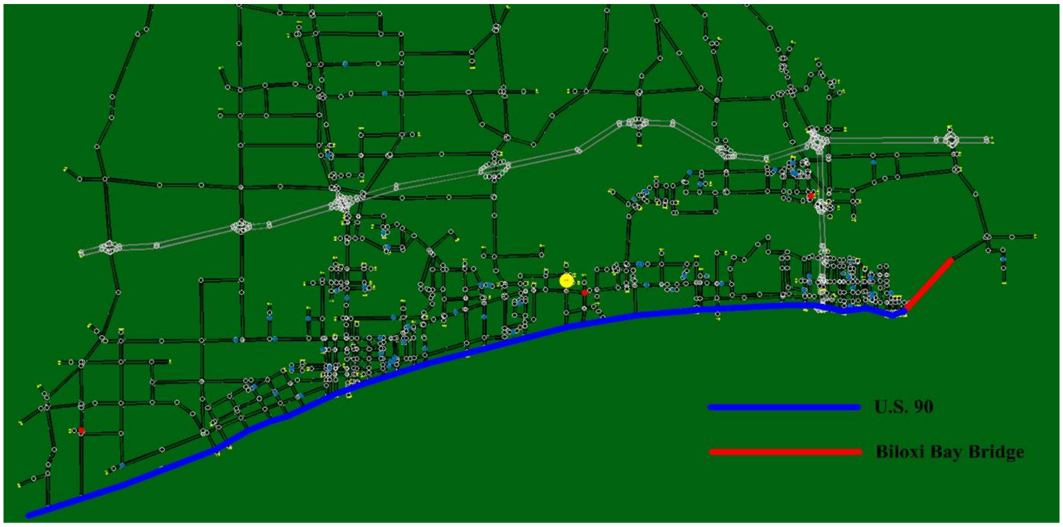

Figure 5.17 CORSIM Network of Scenario 3(b)

The results of scenario 3(b) in t_1 are summarized as follows. The total travel time is 410.6 minutes and the computation time is 177 seconds.

Route 1: cost = 63.4 minutes and load = 26

Dummy Node - Node 27 - Node 5 - Node 18 - Node 8 - Node 14 - Node 7 - Node 22 - Node 15 - Node 19 - Node 9 - Node 4 - Node 70 - Node 1

Route 2: cost = 71.0 minutes and load = 20

Dummy Node - Node 6 - Node 21 - Node 17 - Node 67 - Node 61 - Node 60 - Node 63 - Node 62 - Node 25 - Node 66 - Node 1

Route 3: cost = 15.2 minutes and load = 18

Dummy Node - Node 24 - Node 20 - Node 23 - Node 3

Route 4: cost = 45.1 minutes and load = 29

Dummy Node - Node 59 - Node 47 - Node 36 - Node 68 - Node 32 - Node 48 - Node 37 - Node 3

Route 5: cost = 36.7 minutes and load = 27

Dummy Node - Node 10 - Node 13 - Node 12 - Node 26 - Node 16 - Node 11 - Node 1

Route 6: cost = 110.8 minutes and load = 28

Dummy Node - Node 44 - Node 39 - Node 50 - Node 57 - Node 69 - Node 29 - Node 28 - Node 34 - Node 49 - Node 33 - Node 55 - Node 42 - Node 53 - Node 51 - Node 65 - Node 64 - Node 2

Route cost = 68.4 minutes and load = 30

Dummy Node - Node 41 - Node 31 - Node 35 - Node 40 - Node 52 - Node 30 - Node 46 - Node 58 - Node 45 - Node 43 - Node 54 - Node 56 - Node 38 - Node 2

Figure 5.18 shows the updated transit routes in scenario 3(b) t_3.

Figure 5.18 Updated Transit Routes in Scenario 3(b) t_3

Four routes, route 1, route 2, route 3, and route 5, are revised after the Biloxi Bay Bridge broken. See Figure 5.19 – Figure 5.22 for a comparison of the results between scenario 3(a) and scenario 3(b) in t_1.

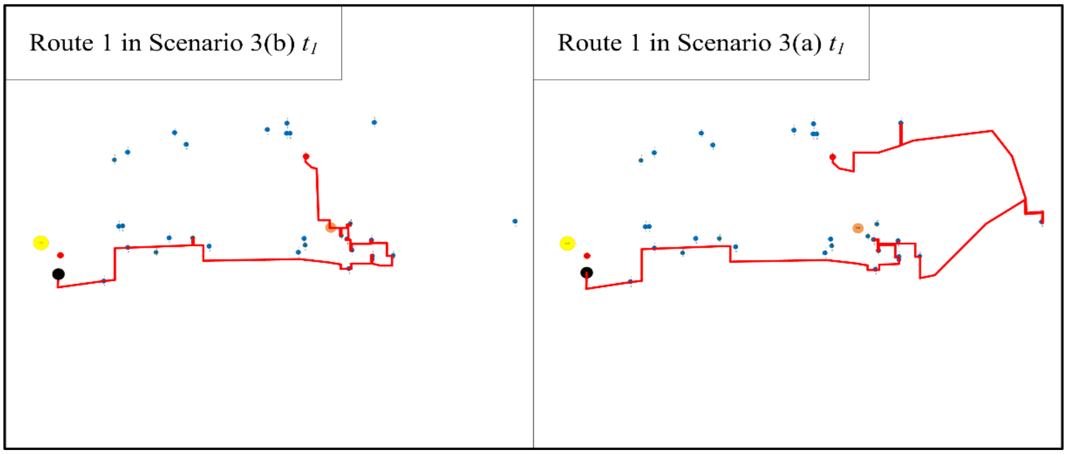

Figure 5.19 Comparison of Route 1 between Scenario 3(a) t_3 and Scenario 3(b) t_3

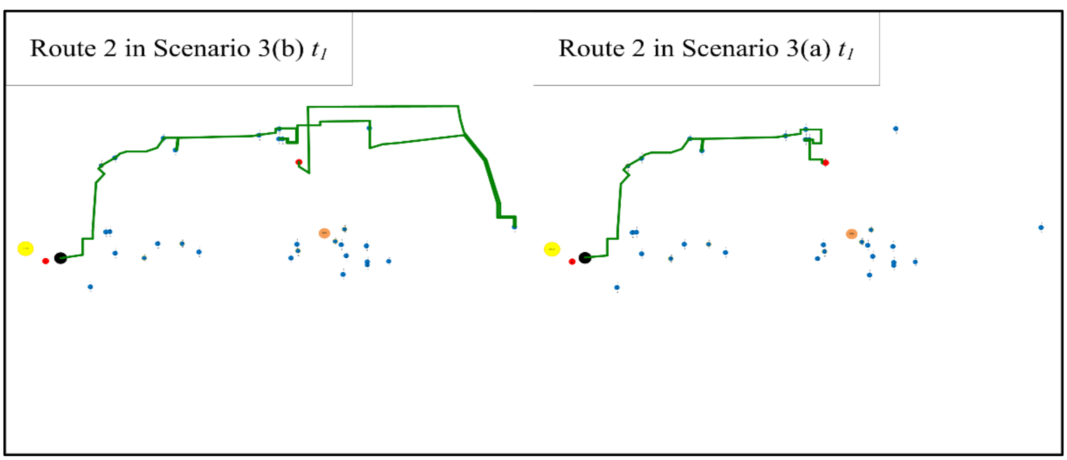

Figure 5.20 Comparison of Route 2 between Scenario 3(a) t_3 and Scenario 3(b) t_3

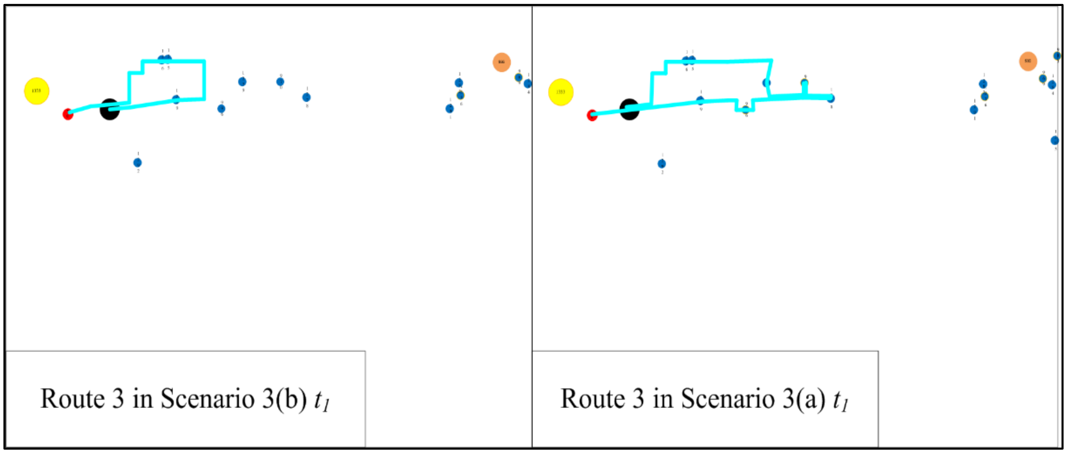

Figure 5.21 Comparison of Route 3 between Scenario 3(a) t_3 and Scenario 3(b) t_3

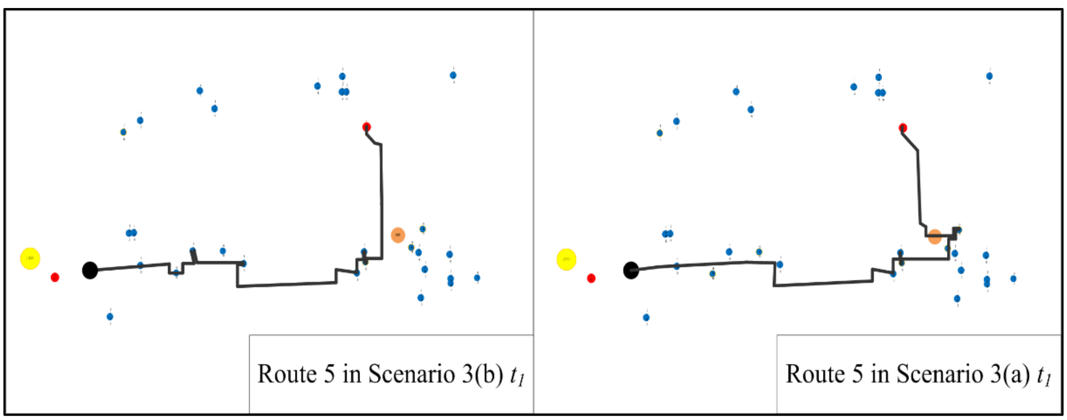

Figure 5.22 Comparison of Route 5 between Scenario 3(a) t_3 and Scenario 3(b) t_3

In scenario 3(b), the Biloxi Bay Bridge is hypothetically broken after the emergency evacuation starts. As a result, route 1 in scenario 3(a) is no longer applicable to node 25 and node 66. After re-optimization, node 25 and node 66 are assigned to route 2 vehicle 2 which is the nearest vehicle capable of picking them up. Because node 25 and node 66 are removed from route 1, vehicle 1 will has sufficient capacity to pick up node 4, node 9, and node 70 which are originally carried by vehicle 5. The pickup points in zone 3 are divided into two groups by the Biloxi Bay Bridge and the U.S. 110 Bridge over the Back Bay. The first group including node 25 and node 66 are assigned to vehicle 2 and the second group including the rest of nodes in zone 3 are covered by vehicle 1. This re-assignment impacts route 3 and route 5 as well. The pickup points along with Pass Road section between Popps Ferry Road and Rodeo Drive are distributed to route 3 and route 5 optimally.

The rest of results from t_3 to t_{30} are listed in Table 5.5.

Table 5.5 Results of Scenario 3(b)

Time Interval	Total Cost (*Minute*)	No. of Vehicle	Computation Time (*Second*)
t_0	417.9	7	161
t_1	410.6	7	177
t_2	394.9	7	180
t_3	365.6	6	156
t_4	337.5	6	135
t_5	314.7	6	54
t_6	301.8	7	50
t_7	260.2	5	47
t_8	251.4	5	44
t_9	237.1	5	31
t_{10}	216.8	5	23
t_{11}	192.5	5	13
t_{12}	172.2	5	9
t_{13}	167.8	4	6
t_{14}	182.5	4	3
t_{15}	169.5	4	3
t_{16}	158.1	3	3
t_{17}	146.4	4	2
t_{18}	147.4	4	2
t_{19}	136.2	4	1
t_{20}	128.9	3	1
t_{21}	132.4	4	1
t_{22}	124.6	4	1
t_{23}	117.1	4	1
t_{24}	104.3	2	1
t_{25}	97.0	2	1
t_{26}	103.1	2	1
t_{27}	94.9	2	1
t_{28}	94.7	2	1
t_{29}	111.5	3	1
t_{30}	120.1	3	1

Scenario 4

In some cases, the transit agency requires to minimize the cumulative time that the evacuees exposes to potential risks. Scenario 4 is developed considering this requirement. The objective is changed to minimize the total time that the evacuees stay in the hurricane affected area before they are evacuated to the shelters. The results are presented as follows.

The total travel time is 426.6 minutes and the computation time is 173 seconds.

Route 1 cost = 43.2, load = 28

Node 0 - Node 23 - Node 10 - Node 13 - Node 5 - Node 12 - Node 20 - Node 24 - Node 3

Route 2 cost = 65.8, load = 20

Node 0 - Node 18 - Node 8 - Node 14 - Node 15 - Node 19 - Node 22 - Node 7 - Node 66 - Node 25 - Node 1

Route 3 cost = 93.3, load = 29

Node 0 - Node 41 - Node 44 - Node 39 - Node 57 - Node 50 - Node 29 - Node 28 - Node 69 - Node 6 - Node 21 - Node 17 - Node 67 - Node 60 - Node 61 - Node 62 - Node 63 - Node 1

Route 4 cost = 72.3, load = 22

Node 0 - Node 32 - Node 34 - Node 49 - Node 33 - Node 55 - Node 58 - Node 43 - Node 54 - Node 56 - Node 38 - Node 64 - Node 2

Route 5 cost = 45.6, load = 29

Node 0 - Node 59 - Node 37 - Node 48 - Node 68 - Node 36 - Node 46 - Node 30 - Node 3

Route 6 cost = 65.6, load = 24

Node 0 - Node 31 - Node 35 - Node 47 - Node 52 - Node 40 - Node 45 - Node 51 - Node 42 - Node 53 - Node 65 - Node 2

Route 7 cost = 40.8, load = 30

Node 0 - Node 27 - Node 26 - Node 11 - Node 16 - Node 9 - Node 4 - Node 1

Figure 5.23 shows the updated transit routes in Scenario 4 t_1.

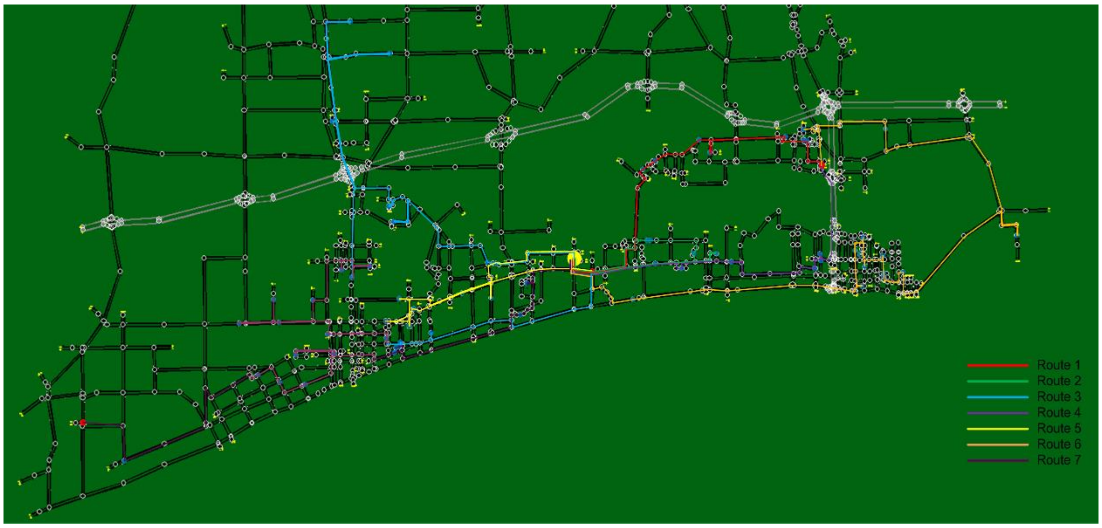

Figure 5.23 Transit Routes in Scenario 4 t_1

Results Analysis

Computation Time

The statistics of the computation time for the three scenarios are displayed in Figure 5.24. First, the SmartEvac system generates an initial solution using 157 seconds for the network with 74 nodes. Then, the computation time increases as the network size grows with new added pickup points and dummy points. The peak computation times are 171 seconds for scenario 2, 178 seconds for scenario 3(a), and 180 seconds for scenario 3(b), which meets the design standard. The average computation times are 28.9 seconds, 34.3 seconds, and 35.9 seconds, in scenario 2, scenario 3(a), and scenario 3(b), respectively. In addition, the computation time shows similar tendency for all the three scenarios that it drops sharply from the 5th interval. The primary reason for this phenomenon is that the network size starts to decrease with the completion of part of the pickup requests. For example, in scenario 2, the network size drops from 76 to 65 in the 5th interval. Another reason is that, as discussed in Chapter III, the initial routes set *R* for each interval is gradually improved with the SmartEvac system running. A high quality initial routes set *R* is able to accelerate the convergence of the column generation algorithm and thus reduce the computation time (Toth et al., 2001).

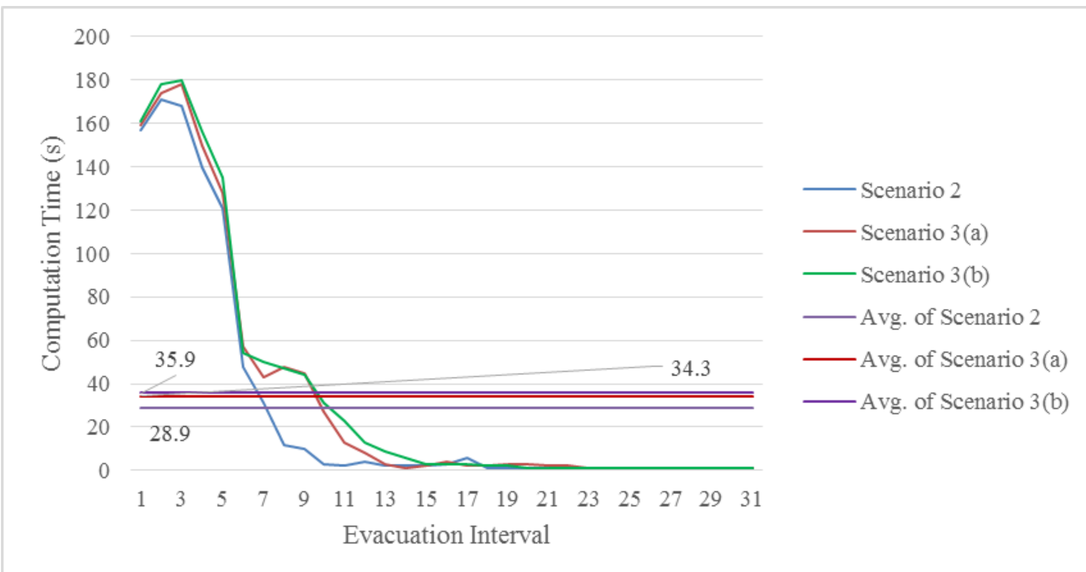

Figure 5.24 Computation Time in Scenario 2, Scenario 3(a), and Scenario 3(b)

Response to Evacuation Information Updates

The ability that the SmartEvac system responses to the dynamic evacuation information, such as new pickup requests, is a primary indicator of the SmartEvac system's applicability in a real time emergency evacuation. The system response time is defined as the interval from a new pickup request coming into the system to the implementation of an updated transit vehicle routing plan considering the new pickup request. It is assumed that the arrival of the new pickup requests are uniformly loaded in the emergency evacuation process. Since the SmartEvac system updates the evacuation information at the end of each interval and re-optimizes the transit vehicle routes in the next interval, the average response time to a new pickup request is $3t/2$ in scenario 2 with fixed time interval, where t is the length of the interval. Therefore, the average response time is related with the length of time interval that the SmartEvac system needs to collect dynamic evacuation information and do re-optimization. However, because of the computation burden at the initial stage of the evacuation process, the fixed interval is usually very lengthy but becomes redundant when the network size decreases to around 60 nodes. In order to overcome the deficiency with fixed time interval, dynamic time interval is applied in scenario 2.

The advantage of dynamic interval is that its length can be dynamically adjusted based on its previous interval's length and computation time. Figure 5.25 draws a comparison of response time between scenario 2 with fixed interval and Scenario 2 with dynamic interval. The response time in scenario 2 with dynamic interval drops continually until the 8^{th} interval after which the response time fluctuates around 100 seconds. The average response time with dynamic interval is 110.8 seconds in contrast with 270.2 seconds with fixed interval.

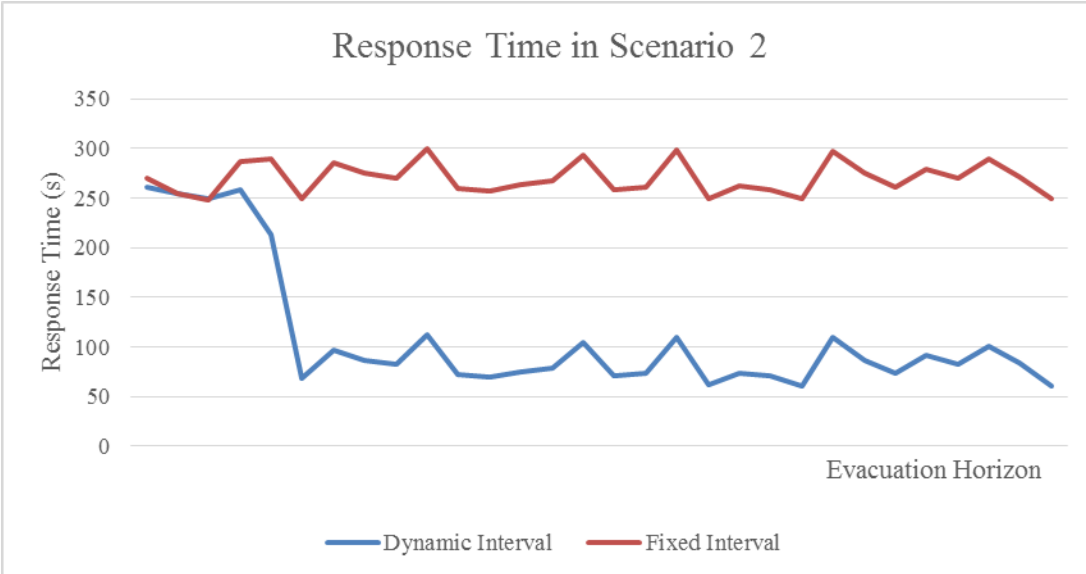

Figure 5.25 Response Time in Scenario 2 with Fixed Interval and Dynamic Interval

Comparison with Real Evacuation Results

An effective way to validate the system is to compare with the results from real Hurricane Gustav evacuation. In Hurricane Gustav evacuation, CTA employed 15 transit vehicles, 15 drivers. There were total 15 vehicle-trips in the whole evacuation process. Table 5.6 summarized the SmartEvac system's results and the results from CTA's record.

Table 5.6 Comparison of SmartEvac system's results and CTA's record

	SmartEvac	CTA Operations	% Saving
Total Evacuation Time (min)	417.9	637.5	34.4
Average Response Time (min)	1.4	10	86.0
No. of Vehicles Used in the Evacuation	7	15	53.3

The results from the SmartEvac system are much more efficient in terms of the total evacuation time, average response time, and number of vehicles used in the evacuation. The total evacuation time improved 34.4% by the SmartEvac system. Most importantly, the SmartEvac system would response to a new pickup request under two minutes while the CTA requires 10 minutes on average. There are only seven vehicles used by the SmartEvac system which is much less than 15 of CTA's usage. All the above results demonstrate that the SmartEvac system significantly outperforms the CTA's old system.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

Conclusions

This dissertation developed a SmartEvac system for real time transit vehicle routing optimization in an emergency evacuation. The objective of the SmartEvac system is to reduce the total travel time of all transit vehicles. In some cases, the objective is revised to minimize the total exposure time of all evacuees.

A column generation based CDVRPPD model is integrated into the SmartEvac system. In a static scheme, the difference between the CDVRPPD model and traditional VRP model is that transit vehicles have to deliver the evacuees to a shelter instead of the depot where they depart. Hence additional constraints are added to the CDVRPPD model for pickup and delivery. In a real-time scheme, the model is reformulated in each interval over the planning horizon. Essentially, the dynamic model can be converted from a multi-depot CDVRPPD to a single-depot CDVRPPD by introducing dummy pickup points. The conversion can obviously reduce the complexity of the CDVRPPD model. Furthermore, dynamic intervals, whose interval length is determined based on the computational performance of last interval, are implemented over the planning horizon. Case study has demonstrated that the response time of the SmartEvac system can be greatly improved by the implementation of dynamic intervals.

The CDVRPPD model is formulated in a set covering form. The set covering model typically contains an exponential number of variables which is impractical to solve it directly. Therefore, a column generation method, which progressively expands the routes set towards the optimum solution instead of enumerating all the routes, is applied to solve the model. The column generation operation is based on a master-problem and sub-problem structure. The master problem model guides the routes set expansion while the sub-problem model is developed to price out all the routes which are necessary to construct an optimal solution. In a real-time scheme, the initial routes set is generated by integrating Clarke-Wright saving algorithm with insertion heuristic. The routes set from previous interval is revised to be initial routes set of current interval. In order to improve the lower bound, a cycle elimination algorithm is proposed to solve the pricing sub-problem. The cycle elimination algorithm firstly adopts a 2-cycle elimination procedure (Larsen, 1999) to erase the cycles of length 2 in the solution. Then, a structural resource is iteratively imposed to the nodes which form cycles of length > 2. This procedure can effectively eliminate the cycles with length > 2 so that the lower bound is significantly improved. The computational results indicate that the average improvement of lower bound reaches 12.5% on the benchmark problems in compare with Agarwal, Mathur, and Salkin (1989), Bixby (1998), and Hadjiconstantinou, Christofides, and Mingozzi (1995)'s results in the literature. In addition, the computation time still locates in an affordable range for a real-time system when dealing with the clustered benchmark problems with network size of 50 – 100. The increase of computational time by introducing cycle elimination reveals that the system is suitable for a network of size around 100.

A case study based on Hurricane Gustav evacuation is proposed to demonstrate the SmartEvac system in real scenarios. CORSIM simulations are developed to provide data for the SmartEvac system. Transportation network data in Gulf Coast area are collected in field survey. CORSIM RTE is developed as an interface to exchange data between CORSIM simulation and the SmartEvac system. Different scenarios corresponding to the different situations that happened in the Hurricane Gustav emergency evacuation are proposed to evaluate the performance of the SmartEvac system in response to real-time data. The average processing time is 28.9 seconds and the maximum processing time is 171 seconds (scenario 2), which demonstrate the SmartEvac system's capability of real-time vehicle routing optimization.

In summary, the major contribution of this dissertation is the development of a SmartEvac system which is able to handle real-time transit vehicle routing in an emergency evacuation of 200 – 250 evacuees. Traditional VRP model is revised to be applicable in a real-time scheme. The implementation of dynamic intervals could effectively reduce the system response time to an emergency. In addition, the proposed cycle elimination algorithm could tight the lower bound without contaminating the overall performance of the system.

Future Work

Related studies which are not in the scope of this dissertation will be conducted in the future. The followings summarized the future work.

1. Parallel computing. In this dissertation, the column generation algorithm can handle problems up to about 100 evacuees. For larger problems, parallel computing, which increases the computation power, can be a good method. To

implement parallel computing, an efficient parallel algorithm is necessary. Potential implementation of parallel computing is to use two phase algorithm, where the evacuees are clustered first, and then routing is performed in each cluster. In this scheme, a master computer is in charge of clustering and other parallel computers are in charge of routing in each cluster. Another potential implementation of parallel computing is to introduce the branching and bounding procedure when solve the master problem as an integer problem. The master computer is able to handle the termination of the algorithm while other parallel computers solve at different branch and bound nodes. In the past, results from Larsen (1999) and Clausen (1999) showed that parallel computing can efficiently solve CDVRP up to 500 customers with proper implementation.

2. More Efficient Algorithms. In the future, more efficient algorithms are expected to be implemented in the system. Meta-heuristics, such as tabu search and simulated annealing, have been demonstrated their effectiveness in solving large scale vehicle routing problems. Therefore, the SmartEvac system is expected to implement or combine with certain meta-heuristic algorithm, which will make it more applicable in a large scale evacuation. Although the results from a meta-heuristic algorithm may not be optimum, most of time it is still acceptable under emergency situations.

3. Web-based Interface Development. A web interface for the SmartEvac system is in the scope of future work so that authorized transit agencies can access the service anytime through internet/intranet. A user-friendly interface will be

developed. The service will receive dynamic evacuation information and traffic information updates and generate the transit operation plan in real time. Transit agencies can push the outputs into GIS messages, which can be sent to the transit driver's smart phones with Google Navigation.

4. Travel Time Prediction Improvement. In this dissertation, the travel time for the next interval is predicted using a moving average method. However, the parameter λ is unified throughout the network. This implementation could be improper when the travel time changes drastically in the evacuation. In the future, λ may be assigned different values for different areas. For example, λ is smaller in rural area than in urban area. In addition, the value of λ can be dynamically adjust along with the time.

REFERENCES

Alba, Enrique, and Dorronsoro, Bernabe. "Solving the vehicle routing problem by using cellular genetic algorithms." Evolutionary Computation in Combinatorial Optimization. Springer Berlin Heidelberg, 11-20, 2004.

Agarwal, Y., Mathur, K., and Salkin, H.M. "A set-partitioning-based exact algorithm for the vehicle routing problem." Networks, 19:731-749, 1989.

ARCADIS U.S., Inc. "Evacuation time estimates for browns ferry nuclear power plant plume exposure pathway emergency planning zone." TM120006.0001, November, 2011.

ARCADIS U.S., Inc. "Evacuation time estimates for the three mile island plume exposure pathway emergency planning zone" B0033739.0000, December, 2012.

Balas, Egon. "The prize collecting traveling salesman problem." Networks 19.6: 621-636, 1989.

Balinski, M., and Quandt, R. "On an integer program for a delivery problem." Operations Research, 12:300-304, 1964.

Beasley, J. E., and Nicos Christofides. "An algorithm for the resource constrained shortest path problem." Networks 19.4: 379-394, 1989.

Bellman, Richard. "On a routing problem". Quarterly of Applied Mathematics 16: 87–90. MR 0102435, 1958

Bhaduri, B., C. Liu, and O. Franzese. "Oak Ridge evacuation modeling system (OREMS): A PC-based computer tool for emergency evacuation planning." Symposium on GIS for Transportation, 2006.

Bixby, A. "Polyhedral analysis and effective algorithms for the capacitated vehicle routing problem." Ph.D. Dissertation, Northwestern University, Evanston, IL, 1998.

Boyd, Stephen, and Vandenberghe, Lieven. "Convex optimization." Cambridge university press, 2009.

Chen, Zhi-Long, and Hang Xu. "Dynamic column generation for dynamic vehicle routing with time windows." Transportation Science 40.1: 74-88, 2006.

Chiu, Yi-Chang, Pavan Korada, and Pitu B. Mirchandani. "Dynamic traffic management for evacuation." Proc. of the 84th Annual Meeting of the Transportation Research Board (CD-ROM), Washington DC, 2005.

Chiu, Y.-C., H. Z. Hong, et al. "Evaluating regional contra-flow and phased evacuation strategies for Texas using a large-scale dynamic traffic simulation and assignment approach." Journal of Homeland Security and Emergency Management 5(1), 2008.

Chiu, Y. C., et al. "DynusT User's Manual." Available: http://dynust.net/, 2010

Christofides, N. and Eilon, S. "An algorithm for the vehicle dispatching problem." Operational Research Quarterly, 20:309-318, 1969.

Christofides, N., Mingozzi, A., and Toth, P. "The vehicle routing problem." In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, Combinatorial Optimization, Wiley, Chichester, UK, pp. 315-338, 1979.

Citilabs. "Cube voyager operating manual." Tallahassee FL USA, 2013.

Clarke, G., and Wright, J.V. "Scheduling of vehicles from a central depot to a number of delivery points." Operations Research, 12:568-581, 1964.

Clausen, Jens. "Branch and bound algorithms-principles and examples." Department of Computer Science, University of Copenhagen: 1-30, 1999.

Cova, Thomas J., Johnson, Justin P. "A network flow model for lane-based evacuation routing." Transportation Research Part A: Policy and Practice, Volume 37, Issue 7, Pp. 579-604, August 2003.

Creput, Jean-Charles, et al. "Dynamic vehicle routing problem for medical emergency management." Self-Organizing Maps-Applications and Novel Algorithm Design, 2011.

Dantzig, G., and Ramser, J. "The truck dispatching problem." Management Science, 6(1): 80-91, 1959.

Dantzig, George B., and Philip Wolfe. "Decomposition principle for linear programs." Operations research 8.1: 101-111, 1960.

Dell'Amico, Mauro, Francesco Maffioli, and Peter Varbrand. "On prize-collecting tours and the asymmetric travelling salesman problem." International Transactions in Operational Research 2.3: 297-308, 1995.

Desrochers, M. "An algorithm for the shortest path problem with resource constraints." Technical Report G-88-27, GERAD, 1988.

Desrochers, M., Desrosiers, J., and Solomon, M. “A new optimization algorithm for the vehicle routing problem with time windows. Operations Research, Volume 40, Number 2, Page 342-354, April 1992.

Dial, R. B. “A probabilistic multipath traffic assignment model which Obviates Path Enumeration.” Transportation Research 5: 83-111, 1971.

Dror, M. “Note on the complexity of the shortest path models for column generation in VRPTW.” Operations Research 42:977-978, 1994.

Edara, Praveen, Siddharth Sharma, and Catherine McGhee. "Development of a large-scale traffic simulation model for hurricane evacuation methodology and lessons learned." Natural Hazards Review 11.4: 127-139, 2010.

Feillet, D., Dejax, P., Gendreau, M., Gueguen. C. “An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems.” Networks, Volume 44, Issue 3, Pages 216–229, October 2004.

Feillet, Dominique, Pierre Dejax, and Michel Gendreau. "Traveling salesman problems with profits." Transportation science 39.2: 188-205, 2005.

FEMA. (August 28, 2013). HURREVAC 2013. Retrieved 2014, from http://www.hurrevac.com/.

Ford Jr., Lester R. “Network flow theory.” Paper P-923. Santa Monica, California: RAND Corporation, August 14, 1956.

Goel, A. and Gruhn, V. "Solving a dynamic real-life vehicle routing problem." Operations Research Proceedings 2005: 367-372, 2005.

Golden, Bruce L., Larry Levy, and Rakesh Vohra. "The orienteering problem." Naval research logistics 34.3: 307-318, 1987.

Golden, Bruce L., Wasil, E.A., Kelly, J.P., et al. "The impact of meta-heuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results." Fleet management and logistics. Springer US, 33-56, 1998.

Hadjiconstantinou, E., Christofides, N., and Mingozzi, A. “A new exact algorithm for the vehicle routing problem based on *q*-paths and *k*-shortest paths relaxations.” Annals of Operations Research, 61:21-43, 1995.

Han, Lee D., and Fang Yuan. "Evacuation modeling and operations using dynamic traffic assignment and most desirable destination approaches." 84th Annual Meeting of the Transportation Research Board, Washington, DC. 2005.

Hobeika, A.G., and B. Jamei. “MASSVAC: a model for calculating evacuation times under natural disaster.” Emergency Planning, Simulation Series, Vol. 15, pp. 23-28, 1985.

Hobeika, A.G., and Chang Kyun Kim. “Comparison of traffic assignments in evacuation modeling”, IEEE Transactions on Engineering Management, Vol. 45, No. 2, May 1998.

Houck, D.J., J.C. Picard, M. Queyranne, R.R. Vemuganti. “The travelling salesman problem as a constrained shortest path problem: theory and computational experience.” Operation Research 17, 93-109, 1980.

Hunter, J. Stuart. "The exponentially weighted moving average." J. QUALITY TECHNOL. 18.4: 203-210, 1986.

Ichoua, Soumia, Michel Gendreau, and Jean-Yves Potvin. "Planned route optimization for real-time vehicle routing." Dynamic Fleet Management. Springer US. 1-18, 2007.

Irnich, Stefan, and Daniel Villeneuve. "The shortest-path problem with resource constraints and k-cycle elimination for k⩾ 3." INFORMS Journal on Computing 18.3: 391-406, 2006.

Jha, M., Moore, K. and Pashaie, B. “Emergency evacuation planning with microscopic traffic simulation.” Transportation Research Record, No. 1886, 40-48, 2004.

Jin, Mingzhou, Kai Liu, and Burak Eksioglu. "A column generation approach for the split delivery vehicle routing problem." Operations Research Letters 36.2: 265-270, 2008.

Kim, Sangho, Shashi Shekhar, and Manki Min. "Contraflow transportation network reconfiguration for evacuation route planning." Knowledge and Data Engineering, IEEE Transactions on 20.8: 1115-1129, 2008.

KLD Associates INC. “Formulations of the DYNEV and IDYNEV traffic simulation models used in EESF.” Rep. Prepared for the Federal Emergency Management Agency, Commack, NY, 1984.

Kwon, Eil. “Development of operational strategies for travel time estimation and emergency evacuation on a freeway network.” No. MN/RC-2004-49, 2004.

Larsen, Jesper. “Parallelization of the vehicle routing problem with time windows.” Dissertation. Technical University of Denmark, Department of Informatics and Mathematical Modeling, 1999.

Larsen, Allan. “The dynamic vehicle routing problem.” Dissertation. Technical University of Denmark, Department of Transportation, Logistics & ITS, 2000.

Lim, Yu Yik. “Modeling and evaluating evacuation contraflow termination point designs.” Master's Thesis. Louisiana State University, 2003.

Liu, Ying, Xiaorong Lai, and Gang-Len Chang. "Two-level integrated optimization system for planning of emergency evacuation." Journal of transportation Engineering 132.10: 800-807, 2006.

Lund, Karsten, Oli B. G. Madsen, and Jens M. Rygaard. “Vehicle routing problems with varying degrees of dynamism.” Lyngby: IMM Institute of Mathematical Modelling, 1996.

Mahmassani, Hani S., Hu, Ta-Yin, Peeta S., and A. Ziliaskopoulos. “Development and testing of dynamic traffic assignment and simulation procedures for atis/atms applications.” Report DTFH61-90-R-00074-FG, U.S. DOT, Federal Highway Administration, McLean, Virginia, 1994.

Mahmassani, Hani S., Hayssam Sbayti, and Xuesong Zhou, “DYNASMART-P, intelligent transportation network planning tool, version 1.0 user’s guide.”, Maryland Transportation Initiative, University of Maryland, College Park, September 2004.

McTrans. McTrans Center » University of Florida. [Online] Mctrans.ce.ufl.edu. Available at: http://mctrans.ce.ufl.edu/mct/ [Accessed 28 Apr. 2014].

Nagel, Kai, et al. "TRANSIMS traffic flow characteristics." Los Alamos National Laboratory. USDOT/USEPA. Preprint Jun 1997.

Naghawi Hana. “Transit-based emergency evacuation modeling with microscopic simulation.” Louisiana State University and Agricultural and Mechanical College, Department of Civil and Environmental Engineering, 2010.

Naser, Mohammad, and Shawn C. Birst. “Mesoscopic evacuation modeling for small-to medium-sized metropolitan areas.” No. MPC Report No. 10-222. Mountain-Plains Consortium, 2010.

Nemhauser, George L., and Laurence A. Wolsey. “Integer and combinatorial optimization.” Vol. 18. New York: Wiley, 1988.

Peeta, Srinivas, Sushant Sharma, and Yu-Ting Hsu. "Dynamic real-time routing for evacuation response planning and execution." 2011.

Pradhan, Anu, and G. Mahinthakumar. "Finding all-pairs shortest path for a large-scale transportation network using parallel floyd-warshall and parallel Dijkstra algorithms." Journal of Computing in Civil Engineering 27.3: 263-273, 2012.

PTV, AG. "VISSIM 5.20 user manual." Karlsruhe, Germany, 2009.

Raiyn, Jamal, and Tomer Toledo. "Real-time short-term forecasting based on information management." Journal of Transportation Technologies 4: 11, 2014.

Righini, G., Salani, M. “New dynamic programming algorithms for the resource-constrained elementary shortest path problem.” Networks 51.3: 155-170, 2008.

Schomborg, A., K. Nökel, and A. Seyfried. "Evacuation assistance for a sports arena using a macroscopic network model." Pedestrian and Evacuation Dynamics. Springer US. Pp. 389-398, 2011.

Schwardt, M. and J. Dethloff. "Solving a continuous location-routing problem by use of a self-organizing map." International Journal of Physical Distribution and Logistics Management 35(6): 390, 2005.

Sheffi, Y., Mahmassani, H., and Powell, W.B, “A transportation network evacuation model.” Transportation Research, Part A, 16(3), pp. 209-218, 1982.

Songchitruksa, Praprut, et al. "Dynamic traffic assignment evaluation of hurricane evacuation strategies for the Houston-Galveston, Texas, region." Transportation Research Record: Journal of the Transportation Research Board 2312.1: 108-119, 2012.

Stepanov, Alexander, and James MacGregor Smith. "Multi-objective evacuation routing in transportation networks." European Journal of Operational Research 198.2: 435-446, 2009.

Tagliaferri, Anthony Paul. "Use and comparison of traffic simulation models in the analysis of emergency evacuation conditions." Master's Thesis. North Carolina State University, 2005.

Theodoulou, Gregoris, and Brian Wolshon. "Alternative methods to increase the effectiveness of freeway contraflow evacuation." Transportation Research Record: Journal of the Transportation Research Board 1865.1: 48-56, 2004.

Toth, Paolo, and Daniele Vigo. “The vehicle routing problem.” Society for Industrial and Applied Mathematics, 2001.

Urbanik, I. I., and A. E. Desrosiers. “An analysis of evacuation time estimates around 52 nuclear power plant sites analysis and evaluation.” No. PNL-3662 Vol. 1; NUREG/CR-1856. Pacific Northwest National Laboratory (PNNL), Richland, WA, 1981.

USDOT. “Report to congress on catastrophic hurricane evacuation plan evaluation.” In cooperation with the U.S. Department of Homeland Security. Washington, D.C., June 2006.

Wang, Feng, et al. "Study of the effects of evacuation routes and traffic management strategies in short-notice emergency evacuation in downtown Jackson." Transportation Research Board 93rd Annual Meeting, No. 14-4920, 2014.

White, Richard A., et al. "The role of transit in emergency evacuation." Transportation Research Board, Washington DC, 2008.

Williams, B. M., Tagliaferri, A. P., Meinhold, S. S., Hummer, J. E., and Rouphail, N. M. "Simulation and analysis of freeway lane reversal for coastal hurricane evacuation." J. Urban Planning, 133(1), 61–72, 2007.

Wen, Y., Zhang, L., and Huang, Z. "Coordination of connected vehicle and transit signal priority in transit evacuations." Transportation Research Board Annual Meeting, Paper #12-3152, 2012.

Wolshon, Brian, et al. "Application of TRANSIMS for the multimodal micro-scale simulation of the New Orleans emergency evacuation plan final report." 2009.

Wolshon, Brian, and Vinayak V. Dixit. "Traffic modelling and simulation for regional multimodal evacuation analysis." International Journal of Advanced Intelligence Paradigms 4.1: 71-82, 2012.

Yuan, F., L. D. Han, S. Chin., and H. Hwang. "Proposed framework for simultaneous optimization of evacuation traffic destination and route assignment" Journal of the Transportation Research Board, No. 1964, Transportation Research Board of the National Academies, Washington DC, pp 50-58, 2006.

Zhang, Li, et al. "Simulation modeling and application with emergency vehicle presence in CORSIM." Vehicular Technology Conference Fall (VTC 2009-Fall), IEEE 70th. IEEE, 2009.

Zhang, Li, Yi Wen, and Minzhou Jin. "The framework for calculating the measure of resilience for intermodal transportation systems." No. NCIT Report# 10-05-09. 2009.

Zheng, Hong, et al. "Modeling of evacuation and background traffic for optimal zone-based vehicle evacuation strategy." Transportation Research Record: Journal of the Transportation Research Board 2196.1: 65-74, 2010.

Ziliaskopoulous, A.K., Mahmassani, H.S. "A note on least time path computation considering delays and prohibitions for intersection movements." Transportation Research Part B 30, 359–367, 1996.

Zou, Nan, et al. "Simulation-based emergency evacuation system for Ocean City, Maryland, during hurricanes." Transportation Research Record: Journal of the Transportation Research Board 1922.1: 138-148, 2005.