JOURNAL OF ADVANCED TRANSPORTATION
J. Adv. Transp. 2014; **48**:734–749
Published online 24 February 2013 in Wiley Online Library (wileyonlinelibrary.com). **DOI**: 10.1002/atr.1224

Optimization of bus routing strategies for evacuation

Hong Zheng*

Department of Civil Engineering and Engineering Mechanics, The University of Arizona, Tucson, AZ 85721, U.S.A.

SUMMARY

Efficient transportation of evacuees during an emergency has long been recognized as a challenging issue. This paper investigates emergency evacuation strategies that rely on public transit, where buses run continuously, rather than fixed route, based upon the spatial and temporal information of evacuee needs. We formulated an optimal bus operating strategy that minimizes the exposed casualty time rather than operational cost, as a deterministic mixed-integer program, and investigated the solution algorithm. A Lagrangian-relaxation-based solution algorithm was developed for the proposed model. Numerical experiments with different problem sizes were conducted to evaluate the method. Copyright © 2013 John Wiley & Sons, Ltd.

KEY WORDS: evacuation; public transit; Lagrangian relaxation

1. INTRODUCTION

Efficient transportation of evacuees during emergencies has long been recognized by both academia and practitioners as a challenging issue. Significant contributions to optimal evacuation strategies have been made emphasizing on evacuating *automobiles* [1–3]. Unfortunately, in tactical automobile management plans, population involved in a metropolitan area may easily surpass the maximum roadway service capabilities. Therefore, many effective auto-evacuation plans achieve the evacuation goal but with a long clearance time (usually several hours), which is undesirable to prevent serious casualties [4].

This paper investigates evacuation strategies that rely on *public transit*. Mass transit utilization may significantly reduce the congestion resulting from an overwhelming number of passenger cars and may facilitate a more timely evacuation [5,6]. In the case of a no-notice disaster—a disaster that happens in a sudden manner with little or no warning to management agencies [7]—evacuees may not be able to access their vehicles. In such a case, public transit could play a vital role in evacuation.

Several deterministic analytical transit evacuation models formulated as an extension of the vehicle routing problem (VRP) were found in literature [8–10]. As such models are NP-hard in general, most pieces of work in the current practice address model formulation extensively, but bring either simple heuristic rules, or simulation-based methods, to solve the model. For instance, Pages *et al*. [11] proposed a route-based formulation for a mass transportation network design problem, in which a set of routes were determined with a goal of minimizing passengers' in-vehicle time plus waiting time. Their model had a structure similar to that of the network design problem; an iterative heuristic approach was developed to solve it. Sbayti and Mahmassani [12] developed a system-optimal dynamic traffic assignment framework for the purpose of scheduling trips to a set of safe destinations with a minimal system travel time. A simulator, DYNASMART-P (Federal Highway Administration, Washington, DC, USA), was applied in their solution approach. He *et al*. [13] developed an optimization model to generate

*Correspondence to: Hong Zheng, Department of Civil Engineering and Engineering Mechanics, The University of Arizona, Tucson, AZ 85721. E-mail: hzheng@email.arizona.edu

evacuation plans for transit-dependent residents accounting for demand uncertainty. Their model was solved by a genetic algorithm in a random search manner. Abdelgawad *et al.* [8] investigated a transit routing strategy with time constraints by extending a multi-depot VRP. Several objective functions, such as to minimize routing costs, vehicle costs, and waiting time, were investigated. Heuristic rules in a constraint programming approach based on ILOG-dispatcher were used to solve the problems. Recently, Sayyady and Eksioglu [9] studied the optimal usage of public transit in a no-notice evacuation scenario. They formulated the problem as a mixed-integer problem and applied a Tabu search heuristic to solve the problem. It is noticed that all aforementioned analytical models [9,10] applied heuristics in their solution procedures and thus with unknown solution qualities. Differentiating from those studies, we study the solution algorithm in depth. In addition to the analytical approaches, several researchers have applied simulation-based methods to explore the effectiveness of operating public transit for evacuation in urban areas [14–17].

Compared with the classical VRP, a set of practice-oriented constraints arises from applications of public transit in realm, which makes a transit evacuation model different from the VRP in the following manners: (i) It is more difficult to match the time profile of the demand. The demand of evacuees arriving at bus stops is time varying and depends on the walking/running time needed to access a nearby bus stop. (ii) Evacuees waiting at a bus stop are subject to the maximum waiting time. Evacuees may give up on the service if it comes with a long waiting time. (iii) Because of the limited fleet size, buses are expected to serve multiple runs continuously. Overall, a transit evacuation model to a certain degree is similar to the VRP with time windows (VRPTW), for evacuees waiting at a bus stop must be served within an allowable time window. Other model features in the category of VRP that are also involved in a transit evacuation model include multiple depots and open routes (i.e., buses that completed all of the service needs may pull in to a different depot other than where they pulled out from).

The VRP defines a family of problems, which finds the least-cost service routes from a depot to a set of geographically scattered customers, subject to a set of side constraints. The VRPTW is a variant of the classical VRP, where each customer must be visited within a specified time interval, called a time window. The most well-seen algorithmic technique to VRPTW is the column generation algorithm [18], in which the linear programming relaxation of the set of partitioning formulations of the VRPTW is solved by column generation. Feasible columns are added as needed by solving a shortest-path problem with time windows. The obtained linear programming solution is then fed into a branch-and-bound algorithm to solve the integer set partitioning formulation. The algorithm has been reported to successfully solve a 100-customer problem. Another common algorithmic technique is based on Lagrangian relaxation [19], where the constraint requiring that each customer must be served by one vehicle once is relaxed, and it yields the similar constrained shortest-path sub-problem with time windows as the column generation. However, the constrained shortest-path problem with time windows (SPPTW) is NP-hard [20]. In this paper, we apply a similar Lagrangian relaxation method to our transit evacuation model; however, we show that our sub-problem is not characterized by the SPPTW structure.

In this research, we investigate how to efficiently operate available transits for mass evacuation. The motivation of the model is for offline planning purpose, or what-if scenario analysis, rather than online operational analysis, because of several strong assumptions (Section 2.1). We formulate the transit evacuation model as a mixed-integer problem and investigate the solution technique to solve the model. The solution method applies Lagrangian relaxation, and we show that the sub-problem can be transformed into a min-cost flow problem on an extended network involving negative costs but no negative cycles. The min-cost flow problem possesses the integrality property, so our approach is not integrated with the branch-and-bound framework. A construction-based heuristic method was developed to provide an upper bound (UB) in the Lagrangian framework. Contributions of the research are summarized as follows: (i) We propose a deterministic optimization model that operates buses to respond temporally and spatially elaborated evacuees during an emergency. (ii) The model is formulated as a mixed-integer programming with an objective to optimize evacuees' exposed casualties, rather than the operational cost. In addition, buses run continuously, rather than fixed route, on the basis of the where-and-when information of evacuee needs. (iii) We investigate the formulated model structure and propose a specially designed algorithm. (iv) Numerical experiments with different problem sizes are conducted to evaluate the method.

DOI: 10.1002/atr

The remainder of the paper is organized as follows. Section 2 describes the model assumptions and sets up the problem. Mathematical formulations are presented in Section 3, followed by the solution algorithm discussed in Section 4. Section 5 discusses the experiment results of the numerical analysis. Conclusions are given in the last section.

2. THE PROBLEM

2.1. Overview of the problem and assumptions

We assume that a significant number of evacuees opt to take public transit to leave a disaster scene. Given a set of potential pickup stations—usually the same bus stops used in the routine service from a practical operational perspective—the threatened population is guided immediately to the nearest bus stops to call for pickup services. Suppose there is a fleet of empty buses, either waiting at depots or scattered in various locations, that is available to use in the evacuation. Buses are dispatched responding to the pickup requests called in by evacuees and take the evacuees to either a rail stop, where evacuees take a train to continue evacuation, or to an alternate pre-defined delivery/drop-off point, such as a capacitated shelter. The goal is to complete the entire bus dispatch procedure in the quickest possible manner to minimize the risk exposed to evacuees.

The following summarizes several evacuation-oriented components addressed in the model:

(1) Demand (evacuees) waiting at a bus stop is time dependent and temporally elaborated. Most prior research [8,13,9] has assumed that evacuees are ready at a bus stop as soon as the disaster occurs, which is an oversimplified case. In practice, evacuees need time to go to the stop even if responding to the threat immediately. Therefore, evacuees arriving at a bus stop form a temporally continuous stream. We consider the time dependence of demand at a bus stop in the model.
(2) We set a limit on the amount of time that evacuees can afford waiting at a bus stop because with long waiting time, evacuees may give up on the service, particularly in an emergency situation.
(3) The bus routing strategy is integrated with rail lines, if a reliable train service is available. That is, in addition to safe destinations and intermediate shelters, rail stations can also be used as delivery points for buses.
(4) The goal of the transit evacuation model is to minimize the total casualty time experienced by evacuees without concern for the buses' routing cost. We define the casualty time as follows: whenever a bus v completes a drop-off j at a time $T(j)^v$, the passengers in the bus have experienced an amount of time $T(j)^v$ involving dangerous exposure. In such a case, the casualty time for a passenger in bus v is $T(j)^v$. Our goal is to minimize the sum of casualty time experienced by all evacuees, that is, min $\sum_v \sum_j T(j)^v$.

The following list summarizes the assumptions of the model:

(1) The bus fleet size is given.
(2) The geographical location of each bus when a threat occurs is given.
(3) The sizes of buses are identical.
(4) Except for the loading/unloading time at pickups/drop-offs, buses are assumed to be operating continuously. That is, refuel times, crew-change times, and other lost times are ignored, for simplicity.
(5) Assume that en route travel time is static and given because of the limited effective ways to express time-dependent travel time, to maintain the model's tractability at the same time.
(6) The arrival times of the demand at a bus stop are given.
(7) A set of drop-off locations is given.
(8) Evacuees are considered safe whenever they are dropped off, regardless of whether they are dropped off at a train station, or a shelter.

2.2. Discretization of time-dependent demand

To model the time-dependent demand, we discretize the time horizon into small time intervals, indexed by τ, and discretize the demand at a bus stop into a set of individual *pickup requests*. A pickup request is deemed as the smallest unit of demand the model deals with, and one bus stop may involve multiple

DOI: 10.1002/atr

pickup requests. More specifically, let d_w^τ be time-dependent demand at stop w during time interval τ; we discretize d_w^τ into multiple individual *pickup requests* subject to the bus capacity constraint. For example, if $d_w^\tau = 120$ and the bus capacity has $B = 50$, we then divide 120 evacuees into two individual pickup requests, each of 50 evacuees, which equals the bus capacity. As for the remaining 20 evacuees, it is either formed as another individual pickup request, if there are no remaining evacuees at nearby stops, or combined with remaining evacuees at nearby bus stops to constitute a composite pickup request, such that the total evacuees in the composite pickup request is no more than B. This approach has its limitation; if $d_w^\tau \ll B$, then constructing a composite pickup request *per se* becomes a single VRP, which is NP-hard. However, we assume that a disaster event occurs in a metropolis and a significant number of evacuees choose to take public transit to leave. We argue that it is fairly reasonable to assume that $d_w^\tau \gg B$. Therefore, the set of pickup requests is composed primarily of individual pickup requests with only a small portion of composite pickups; the resultant error is negligible. This implies that whenever a bus visits a pickup request, it is filled to capacity and has to proceed immediately to a delivery point to make a drop-off. Operating the bus in such a manner is consistent with the intuitive observation that, if there are a large number of evacuees, whenever a bus stops at a station, it gets fully loaded immediately.

2.3. Constructing the problem on a graph

Let R be a set of pickup requests indexed by i, and each i is associated with a time window $[a_i, b_i]$, where a_i indicates the time at which evacuees in i reach the bus stop and are ready to board, and $b_i - a_i$ represents the maximum allowable waiting time. Note that each i associated with the same bus stop may have different time windows so that multiple pickup requests with variant time windows form a temporally continuous streaming of demand at a bus stop. i must be visited by a bus within its allowable time window $[a_i, b_i]$. Let E be a set of delivery points. D^+ denotes a set of starting pints where a bus pulled out, either from a depot or a pseudo-depot where the bus is en route when evacuation service is called in. D^- denotes a set of end depots. Buses are required to start from D^+ and terminate at D^-. Let V be a set of homogenous buses, indexed by v. We desire to dispatch buses in the following manner: pull out of D^+, visit one pickup request i within its associated time window $[a_i, b_i]$ and go to a drop-off, come back to visit another pickup request within its time window, then go to drop-off, and so on, until no more pickup requests can be visited within the allowable time window; then, pull in to D^-. Each bus is dispatched in such a manner until all pickups have been served by one bus exactly once. The objective is to minimize the total exposure time sustained by the evacuation population. Figure 1 shows four layers of nodes ($D^+, R, E,$ *and* D^-). The goal is to assign buses starting from D^+, visiting R and E alternately and continuously, and ending at D^-, subject to an objective function.

Two decisions are involved in the transit evacuation model. Firstly, the model will determine which pickup request to serve, responding to the where-and-when information of the pickup requests called in as well as the associated time windows. Secondly, the model will decide which delivery point to go to such that the objective function is optimized.

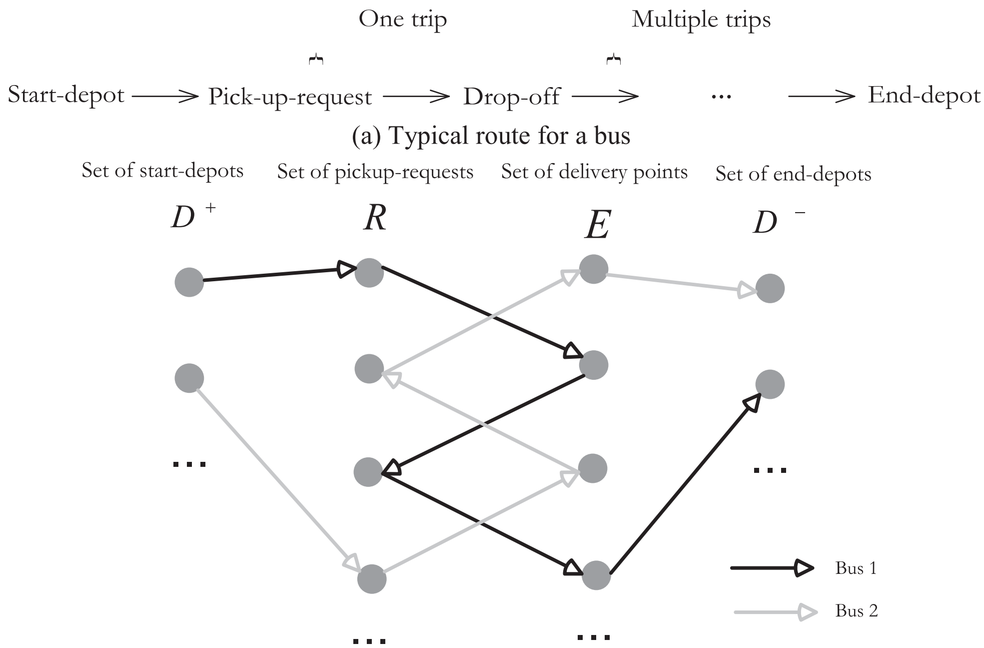
The main differences of the proposed model compared with a family of VRP are as follows: (i) The objective of the model differs from that of the VRP. Because we consider a hard time window, merely minimizing buses' routing cost may lead to undesirable solutions featured with low operational costs but long waiting time. Instead of minimizing the total travel distance (reflective of operational cost), we minimized the total casualty time. (ii) The routing strategy is specific. A bus needs to visit a set of pickup requests and delivery points alternately and continuously, as shown in Figure 1. (iii) There is no fleet capacity constraint in the proposed model because of demand discretization.

3. MATHEMATICAL FORMULATION

Sets:

- D^+ Set of start depots (including pseudo-depots)
- D^- Set of end depots
- D Set of depots, $D = D^+ \cup D^- = \{k | k \in D\}$

DOI: 10.1002/atr



(b) Bus routes in the graph

Figure 1. Graphical layout of bus routes.

R	Set of pickup nodes (i.e., pickup requests)
E	Set of delivery nodes (i.e., delivery points)

Parameters:

$C(k)$	Number of buses available at a start depot $k \in D^+$ or capacity of an end depot $k \in D^-$
t_{ij}	Travel time (a positive number) from node $i \in D^+ \cup R \cup E \cup D^-$ to node $j \in D^+ \cup R \cup E \cup D^-$, $t_{ij} > 0$
$T(k)^v$	Time when bus v departs from a depot $k \in D^+$, $T(k)^v \geqslant 0$
s_i	Service (loading/unloading) time of node $i \in R \cup E$
U_j	Capacity of delivery node $j \in E$, maximal full-bus loads can drop off
M	A constant large number

Variables:

x_{ij}^v	1 if bus v is assigned on arc (i,j), 0 otherwise
$T(i)^v$	Time when bus v visits the pickup node $i \in R$
$T(j)_i^v$	Time when bus v visits the delivery node $j \in E$ after visiting $i \in R$

Model:

$$\min \sum_{v \in V} \sum_{j \in E} \sum_{i \in R} x_{ij}^v \cdot T(j)_i^v \tag{1}$$

s.t.

$$\sum_{v \in V} \sum_{j \in E} x_{ij}^v = 1 \quad \forall i \in R \tag{2}$$

$$\sum_{j \in E} x_{ij}^{v} = \sum_{j \in \{E \cup D^{+}\}} x_{ji}^{v} \quad \forall i \in R, \forall v \in V \tag{3}$$

$$\sum_{i \in \{R \cup D^{-}\}} x_{ji}^{v} = \sum_{i \in R} x_{ij}^{v} \quad \forall j \in E, \forall v \in V \tag{4}$$

$$\sum_{v \in V} \sum_{i \in R} x_{ki}^{v} \leqslant C(k) \ \forall k \in D^{+} \tag{5}$$

$$\sum_{v \in V} \sum_{j \in E} x_{jk}^{v} \leqslant C(k) \ \forall k \in D^{-} \tag{6}$$

$$\sum_{v \in V} \sum_{i \in R} x_{ij}^{v} \leqslant U_j \ \forall j \in E \tag{7}$$

$$T(k)^{v} + t_{ki} \leqslant T(i)^{v} + \left(1 - x_{ki}^{v}\right) M \quad \forall k \in D^{+}, \forall i \in R, \forall v \in V \tag{8}$$

$$T(i)^{v} + s_i + t_{ij} \leqslant T(j)_i^{v} + \left(1 - x_{ij}^{v}\right) M \quad \forall i \in R, \forall j \in E, \forall v \in V \tag{9}$$

$$T(j)_i^{v} + s_j + t_{ji'} \leqslant T(i')^{v} + \left(1 - x_{ji'}^{v}\right) M \quad \forall i, i' \in R, i \neq i', \forall j \in E, \forall v \in V \tag{10}$$

$$a_i \leqslant T(i)^{v} \leqslant b_i \quad \forall i \in R, \forall v \in V \tag{11}$$

$$x_{ij}^{v} \in \{0, 1\} \quad \forall i \in R, \forall j \in E, \forall v \in V \tag{12}$$

Equations (1)–(12) define the transit evacuation model. It is sufficient to define $T(i)^v$ as the unique time when a bus arrives at a pickup node (or pickup request) $i \in R$, but insufficient for a delivery node (or delivery point) $i \in E$, as the same delivery node could be visited by multiple buses several times. Hence, we define $T(j)_i^v$ as the unique time for bus v visiting a delivery point $j \in E$ after service at pickup request $i \in R$. The objective function (1) minimizes the total sum of time when buses visit delivery points, which equals the total casualty time. Constraint (2) indicates that each pickup request must be visited by one bus exactly once. Constraints (3) and (4) maintain flow mass balance at the pickup and delivery nodes, respectively. Constraints (5) and (6) ensure that the total number of buses departing from depot $k \in D^+$ is bounded by the number of available buses and that the total number of buses entering depot $k \in D^-$ is limited by the depot's capacity. Constraint (7) maintains the capacity at delivery points. Constraints (8)–(10) maintain the time window between nodes i and j, if $x_{ij}^v = 1$. Equation (11) ensures that bus v must serve a pickup request within its allowable time window. Finally, Equation (12) maintains integrality. Note that Equations (9) and (10), plus $t_{ij} > 0$, eliminate any sub-tours as the time windows impose a unique route direction for each vehicle. Suppose there is a sub-tour—that is, a pickup request, say i, is visited more than once—one can easily show $T(i)^v < T(i)^v$; this contradicts and thus is impossible.

DOI: 10.1002/atr

Hence, the classical VRP sub-tour elimination constraints become redundant and do not appear in the formula.

Proposition 1. The transit evacuation model formulated in Equations (1)–(12) is strongly NP-hard.

Proof. We show that the transit evacuation model contains an m-identical parallel machine scheduling problem with release time, with an objective function to minimize the total job completion time ($P|r_i|\sum C_i$, following the notation of [21]), which is known to be NP-hard in the strong sense [22].

Construct a restriction of the transit evacuation problem as follows: Let the delivery point followed by each pickup request be fixed and let the travel time from any delivery point to a pickup request be identical. We restrict the time window associated with a pickup request $i \in R$ to be $[a_i, +\infty]$; then, the restriction problem becomes $P|r_i|\sum C_i$, with $r_i = a_i, \forall i \in R$ and $C_i = \sum_{j \in E}\left(x_{ij}^v \cdot T(j)_i^v\right), \forall i \in R$, which is known to be NP-hard. □

4. LAGRANGIAN-RELAXATION-BASED ALGORITHM

4.1. Lagrangian relaxation technique

In this section, we present a solution algorithm based on Lagrangian relaxation. Lagrangian relaxation has been successfully applied in many combinatorial problems including the VRP [19], where the sub-problem is characterized by SPPTW. As our problem structure and objective function differ from those of VRP, we show that the Lagrangian sub-problem has a different structure—a min-cost flow problem with modified cost on an extended network—and therefore, one can solve an integral solution.

For simplicity, we refer to the original problem formulated in Equations (1)–(12) as the primal problem. Dualize the hard constraints (2) and (7); associate dual vector $\lambda_1 = (\lambda_{1i} | i \in R)$ with (2) and $\lambda_2 = (\lambda_{2j} \geqslant 0 | \forall j \in E)$ with (7); the Lagrangian dual problem (LD) is as follows:

$$LD(\lambda) = \max_{\lambda_1, \lambda_2 \geqslant 0} \min \sum_{v \in V} \sum_{j \in E} \sum_{i \in R} x_{ij}^v \left(T(j)_i^v + \lambda_{1i} + \lambda_{2j}\right) - \sum_{i \in R} \lambda_{1i} - \sum_{j \in E} \lambda_{2j} U_j \tag{13}$$

(Equations (3)–(6), (8)–(12))

We intend to solve LD instead of solving the primal problem directly. Given a set of Lagrange multipliers, we call the sub-problem of LD the Lagrangian relaxation problem LR (formulated by Equations (14), (3)–(6), and (8)–(12)). The goal is to solve the Lagrange multiplier that maximizes LD. This can be carried out by a sub-gradient algorithm if LR can be solved efficiently. In the following, we show how to transform LR into a min-cost flow problem on an extended graph, and thus, one can apply a network programming technique to solve it. Additionally, as the network programming possesses the integrality, the approach solves LR with integer numbers and thus will not need to further implement branch-and-bound.

$$LR = \min \sum_{v \in V} \sum_{j \in E} \sum_{i \in R} x_{ij}^v \left(T(j)_i^v + \lambda_{1i} + \lambda_{2j}\right) \tag{14}$$

(Equations (3)–(6), (8)–(12))

4.2. Solving the Lagrangian relaxation problem

We show that LR can be transformed into a min-cost flow problem in a time–space network G^τ. The main idea is to create arcs to represent feasible visits between pickup and delivery nodes and to use arc connectivity to describe feasible movements meeting the constrained time window. Let H be the time horizon. Discretize H into discrete time steps, indexed by τ. For each pickup request $i \in R$, copy node i at the τ time layer if $\tau \in [a_i,b_i]$. Denote the copy of i at the τ time layer by $i(\tau)$. For each delivery node $j \in E$, copy node j at the τ time layer for $\tau \in [0,H]$. Denote the copy of j at the τ time layer by $j(\tau)$. Split node $i(\tau)$ into two nodes, $i(\tau)^+$ and $i(\tau)^-$, and use a directional arc $(i(\tau)^+, i(\tau)^-)$, with cost λ_{1i} and unit capacity, to connect them. Likewise, split node $j(\tau)$ into two nodes $j(\tau)^+$ and $j(\tau)^-$, and use a directional arc $(j(\tau)^+, j(\tau)^-)$, with cost $\tau + \lambda_{2j}$ and infinite capacity, to connect them. For any pair of $i(\tau)^-$ and $j(\tau')^+$ such that $i \in R, j \in E$ and $\tau \leqslant \tau'$, create an arc with zero cost and unit capacity if and only if $\tau + s_i + t_{ij} \leqslant \tau'$ holds, indicating that a bus traversing from pickup node i at time τ to delivery node j at time τ' is feasible. Similarly, for any pair of $j(\tau)^-$ and $i(\tau')^+$ such that $i \in R, j \in E$ and $\tau \leqslant \tau'$, create an arc with zero cost and unit capacity if and only if it satisfies $\tau + s_j + t_{ji} \leqslant \tau'$, indicating that a bus traversing from delivery node j at time τ to pickup node i at time τ' is feasible. For any depot node $k \in D^+ \cup D^-$, split k into two nodes, k^+ and k^-, and create arc (k^+,k^-) in G^τ with zero cost and capacity $C(k)$. For any pair of k^- and $i(\tau)^+$, such that $k \in D^+$, $i \in R$ and $t_{ki} \leqslant \tau$, create arc $(k^-, i(\tau)^+)$ with zero cost and unit capacity, indicating that it is feasible for a bus to pull out from depot k to meet the pickup request i at time τ. Similarly, for any pair of $j(\tau)^-$ and k^+ such that $j \in E$ and $k \in D^-$, create arc $(j(\tau)^-, k^+)$ with zero cost and with capacity $C(k)$. Finally, create a supersink node θ and create arc (k^-,θ) with zero cost and capacity $C(k)$ for each $k \in D^-$. Associate a supply $C(k)$ with each source node $k \in D^+$ and associate a demand $\sum_{k \in D^+} C(k)$ with the supersink node θ. Then, LR has to find the min-cost flow on the constructed time–space network G^τ. Figure 2 illustrates an example of G^τ.

In this example, there are two pickup requests, symbolized by p_1 and p_2, which must be served within the time windows [1,2] and [2,4]. We have one delivery point, denoted by d_1. k_1 denotes the single start depot, and k_2 denotes the single end depot. The travel time between the pickup nodes p_1 and p_2 and the delivery point d_1 is 1. For simplicity, let the travel time from the start depot to the pickup nodes, as well as from the delivery nodes to the end depot, be zero (we should have $t_{ij} > 0$; the one here is for a simple illustration purpose only). The service (loading/unloading) time in this example is also zero. We illustrate the arc connectivity of G^τ in Figure 2.

As the dual variable λ_1 is unrestricted, the cost involved in G^τ may be negative. However, the structure of G^τ indicates that there are no loops involved in the network—it cannot loop over time, and hence, there are no negative cycles contained in G^τ.

The min-cost flow of G^τ provides an optimal solution of LR; it may violate the dualized constraints (2) and (7) in the primal problem. The goal is to find an appropriate dual vector such that

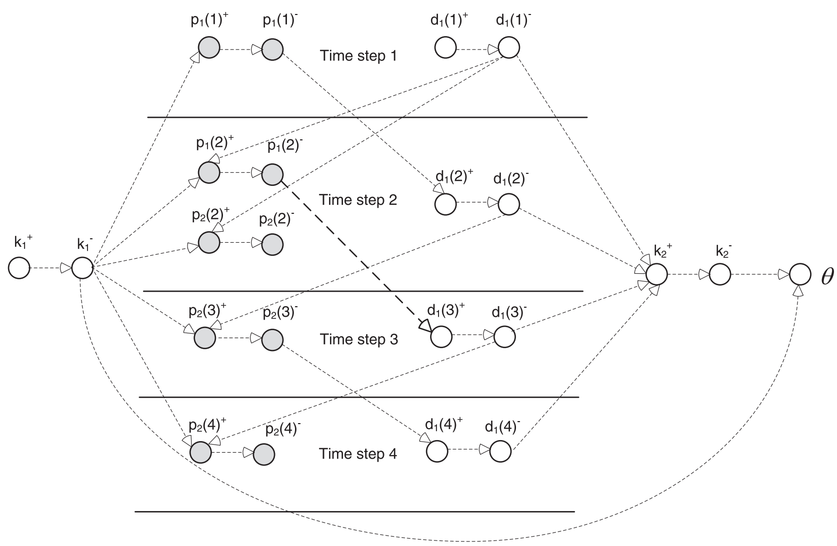


Figure 2. Illustration of extended time–space network G^τ.

$$\sum_{\tau\in[a_i,,b_i]} x\big(i(\tau)^+, i(\tau)^-\big) = 1 \ \text{ for all} \, i \in R \tag{15}$$
$$\sum_{\tau\in[0,H]} x\big(j(\tau)^+, j(\tau)^-\big) \leqslant U_j \ \ for \ \ all j \in E$$
$$\lambda_{2j}\left(\sum_{\tau\in[0,H]} x\big(j(\tau)^+, j(\tau)^-\big) - U_j\right) = 0 \ \ for \ \ all j \in E$$

If $\sum_{\tau\in[a_i,,b_i]} x\big(i(\tau)^+, i(\tau)^-\big) = 0$ (note that the network flow programming solves an integer solution), the Lagrangian approach reduces the cost on arcs $(i(\tau)^+, i(\tau)^-) : i \in R, a_i \leqslant \tau \leqslant b_i$ to encourage carrying flows on those arcs. If $\sum_{\tau\in[a_i,,b_i]} x\big(i(\tau)^+, i(\tau)^-\big) \geqslant 2$, it increases the cost on arcs $(i(\tau)^+, i(\tau)^-)$: $i \in R, a_i \leqslant \tau \leqslant b_i$ as a penalty. The procedure continues until Equation (15) holds for all arcs $(i(\tau)^+, i(\tau)^-) : i \in R$ and $(j(\tau)^+, j(\tau)^-) : j \in E$, and a complementary slackness condition $\lambda_{2j}(\sum_{\tau\in[0,H]} x(j(\tau)^+, j(\tau)^-) - U_j) = 0, \forall j \in E$ is met for the dualized inequality constraint (7). It is noteworthy that one needs to secure feasibility in such an approach; that is, the given number of vehicles is feasible to visit all pickup requests within the allowable time window, otherwise *LR* would never terminate in a sub-gradient algorithm.

It is noted that if we associate a fixed supply/demand function at the sources/sinks, it may not be feasible to simultaneously maintain both a fixed supply/demand function and Equation (15). In such a case, the supply/demand associated with the sources/sinks must be flexible. More specifically, we do not want to specify that $C(k)$ buses at depot k must be in service, as if $C(k)$ is greater than the cardinality of R, then at least one pickup request $i \in R$ violates Equation (15). Instead, we want to say that at most $C(k)$ buses in depot k must be in service. To model the flexible supply/demand, we add a dummy arc to connect node k^- and θ for each start depot $k \in D^+$. Associate a zero cost with arcs $(k^-,\theta) : k \in D^+$ and capacity $C(k)$. Hence, the algorithm would first assign flows on arc (k^-,θ) as much as it can, and then shift flows to arcs $i(\tau)^+, i(\tau)^-$ to maintain Equation (15). Assigning flows on the dummy arc (k^-,θ) implies that $x(k^-,\theta)$ buses are idle at the start depots with no service tasks; the procedure is helpful to identify an optimal solution with the minimal number of fleet.

4.3. Solving the Lagrangian dual problem

The sub-gradient approach has been used to solve many Lagrangian dual problems. It generates a sequence $\lambda^0, \lambda^1, \ldots, \lambda^n$ of Lagrange multiplier vectors heuristically following a descent direction. At each iteration, we check the feasibility and complementary slackness condition of Equation (15). If Equation (15) holds for all pickup requests and delivery nodes, then both the Lagrange multiplier and the primal problems are optimal, and the algorithm terminates. Otherwise, the algorithm generates a new Lagrange multiplier vector, λ^n, by a sub-gradient method. For the iterative technique used to determine successive values of the Lagrange multiplier vector, the choice of step size α^n strongly affects the convergence of *LD*. Typical approaches, for example, $\alpha^n = {}^1/_k$ or ${}^{0.1}\big/_{\sqrt{k}}$, produce long tailing effects before convergence as n becomes large. This occurs because only a very small lower-bound (LB) improvement is obtained because of an excessively small step size. To obtain faster convergence, we apply the heuristic method proposed by Held and Karp [23] to update λ^{n+1}:

$$\lambda^{n+1} = \lambda^n + \alpha^n \frac{UB - LD(\lambda^n)}{\|s(\lambda^n)\|^2} s(\lambda^n) \tag{16}$$

DOI: 10.1002/atr

where UB defines an upper bound to the primal problem, $LD(\lambda^n)$ denotes an optimal solution to LR given λ^n, α^n is a step-size length (a scalar) with $0 \leqslant \alpha^n \leqslant 2$, and $s(\lambda^n)$ defines the sub-gradient direction, which is

$$\begin{aligned} s\left(\lambda_1^n\right) &= \sum_{\tau \in [a_i, b_i]} x\left(i(\tau)^+, i(\tau)^-\right) - 1 \quad \forall i \in R \\ s\left(\lambda_2^n\right) &= \sum_{\tau \in [0,T]} x\left(j(\tau)^+, j(\tau)^-\right) - U_j \quad \forall j \in E \end{aligned} \tag{17}$$

Equation (16) requires a UB. In the next section, we propose a heuristic method that constructs a feasible solution to the primal problem. We choose the step size α^n as follows: Initially, $\alpha^n = 1$, and then half α^n whenever the solved objective function of LR (Lagrangian LB) is less than the best LB that has been obtained for five continuous iterations. Suppose a UB is available, then the sub-gradient algorithm solving the Lagrangian dual problem is presented as follows:

Step 1: Set $n \leftarrow 0$. Initialize a dual vector λ^n. Construct G^τ.
Step 2: Obtain a UB.
Step 3: Update the cost of arc $(i(\tau)^+, i(\tau)^-), \forall i \in R$ as λ_{1i}^n in G^τ and the cost of arc $(j(\tau)^+, j(\tau)^-)$ as $\tau + \lambda_{2j}^n$ in G^τ.
Step 4: Solve the min-cost flow problem on G^τ with the updated cost.
Step 5: Check whether Equation (15) holds for all $i \in R$ and $j \in E$. If "Yes," the algorithm terminates. Both the Lagrangian dual and primal problems are solved to optimal. If "No," compute λ_i^{n+1} by Equations (16) and (17). Set $n \leftarrow n+1$ and go to Step 3.

Because such a step-size rule of α is heuristic, there is a chance that the sub-gradient method cannot solve LD to optimal. In such a case, one can stop the algorithm after a certain run time or after a pre-defined convergence criterion is met. At this stage, the given solution is usually "modestly" infeasible and with only a slightly degradation in the objective function, so one can construct a good feasible solution on the basis of the infeasible solution given by the relaxed problem.

4.4. Obtain an upper bound

In this section, we present a heuristic to obtain a UB quickly for the Lagrangian relaxation algorithm. In the Lagrangian relaxation algorithm, the time horizon needs to be known. The UB method presented in this section also provides one feasible time horizon. This time horizon is then fed to the Lagrangian relaxation algorithm to solve LD.

The typical duty of a bus consists of several trips (several rounds of pickup and drop-off). Thus, we decompose bus routes into trips and construct routes trip by trip. In this iterative procedure, because the last node visited by bus v as well as its visiting time is known in the last iteration, we conduct a look-ahead construction to determine the next pickup (node i) and delivery (node j) so that the sum of times at which service begins at the delivery node j is minimal, that is, $\min \sum_v T(j)^v$. This one-trip construction procedure can be modeled as a min-cost flow on a transformed network, where we use node/arc connectivity to represent valid time window constraints. If a bus's workload consists of at most n trips, then the method undergoes at most n iterations. The approach attributes one trip to each bus in one iteration. To achieve a competitive computational performance, the method presented herein is parallel; that is, it constructs trips to many buses at the same time.

Figure 3 illustrates a one-trip construction. Graph $G = (N,A)$ consists of three types of nodes: (i) Bus nodes with a cardinality of $|V|$. Each node represents a bus v that is associated with the last delivery node that the bus visited, p_v, as well as the time $T(p_v)$ when the service of p_v ended. (ii) Pickup request nodes i that have not been visited. (iii) Delivery nodes j that have a remaining capacity of at least B. We create a supersource node r and a supersink node θ. Then construct the following types of arcs in G: (i) From bus node p_v to pickup request node i, we construct an arc if and only if $T(p_v) + t_{p_v,i} \leqslant b_i$. Associate the arc with the cost $\max\left\{a_i + s_i, T(p_v) + t_{p_v,i} + s_i\right\}$ and unit capacity. (ii) From pickup node i to delivery node j, construct an arc with cost t_{ij} and unit capacity. (iii) From r to bus node p_v, construct

DOI: 10.1002/atr

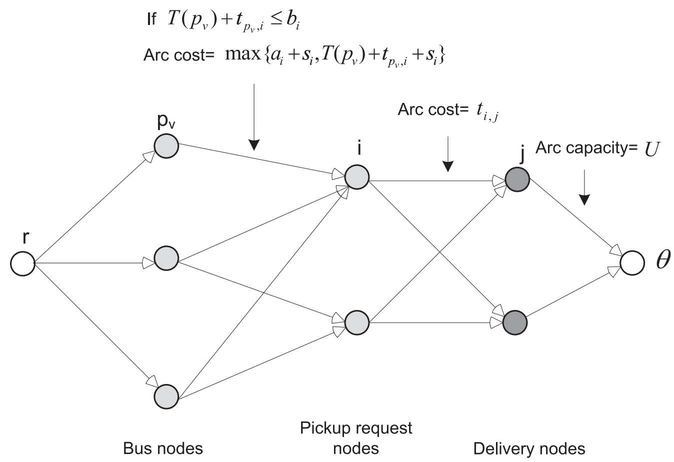


Figure 3. Illustration of G in a one-trip construction.

an arc with zero cost and unit capacity. (iv) From delivery node j to θ, construct an arc with cost s_j and capacity of U, where U denotes the remaining evacuees (in unit of full-bus load, to be consistent with prior discussions) that the delivery node can hold. To assign one trip to each of the buses, we solve the min-cost max (r,θ) flow in G. The heuristic logic is as follows: While there exist some pickup requests that are not being visited, build a graph G and solve the min-cost max (r,θ) flow in G. Finally, let the buses go to the nearest end depots.

As the Lagrangian method needs to maintain feasibility, that is, the given buses must be feasible to visit all pickup requests within allowable time windows, one can quickly run the UB method to see if the given buses are feasible. If infeasible, then it requires more buses added at the start depots and solves the instance again. To obtain the minimal number of buses needed in service, one can apply a binary search framework to run the UB method multiple times.

5. NUMERICAL EXAMPLES

In this section, we conduct numerical analyses to verify correctness and examine the algorithmic performance of the proposed method. The algorithms were implemented in PYTHON 2.5 (Python Software Foundation, Delaware, USA) by using IBM ILOG CPLEX 12.1. The min-cost flow subroutines in both the Lagrangian LB and UB computations were solved by CPLEX. We ran the algorithms on a PC equipped with a 2.33-GHz Intel Core 2 Dual CPU with 2 GB of memory. A small-size illustrative example was applied to verify correctness of the algorithm and compare the solutions produced by different objectives. We used a set of randomly generated real-size examples to compare the solution qualities. The initial vector of Lagrangian multipliers is a set of zeros.

5.1. Small-size example

There are six pickup requests in the example, associated with the time windows tabulated in Table I, and three delivery points. The delivery points are not associated with any of the time windows, for example, there are no time restriction conditions imposed at drop-offs, but with capacity restrictions. Depot information is given in Table II. A matrix displaying the travel time between pickup nodes and delivery points is shown in Table III.

The problem was solved by the Lagrangian relaxation algorithm, and the solution is tabulated in Table IV, with an optimal objective function value of 66. Table IV lists the pickup requests and delivery points visited by each bus in sequence, where the number in brackets indicates the time at which the bus visited the corresponding nodes.

To assess the appropriateness and validity of the proposed objective function, that is, minimizing exposure time, we also compare the solutions produced by different objectives, including minimizing exposure time, minimizing buses' routing time (i.e., operational cost), minimizing buses' waiting time,

DOI: 10.1002/atr

Table I. Parameters given in the example.

Pickup requests (pickup nodes)	Loading time	Time window	Delivery nodes	Unloading time	Capacity (no. of buses)
p1	1	[0,2]	d1	1	2
p2	1	[15,20]	d2	1	4
p3	1	[13,15]	d3	1	4
p4	1	[6,10]			
p5	1	[5,12]			
p6	1	[4,7]			

Table II. Depot information.

Start depot		End depot	
Depot node	Available buses	Depot node	Capacity
D1	2	D3	4
D2	2	D4	4

Table III. Travel time matrix in the example.

Tail\head (head\tail)	p1	p2	p3	p4	p5	p6	D3	D4
d1	5 (6)	3 (1)	1 (10)	14 (5)	10 (7)	14 (9)	1	1
d2	6 (2)	4 (5)	1 (4)	10 (9)	11 (7)	5 (5)	1	1
d3	3 (3)	1 (4)	3 (10)	3 (1)	1 (3)	11 (2)	1	1
D1	2 (8)	1 (1)	15 (9)	3 (8)	13 (4)	11 (10)		
D2	14 (9)	1 (3)	12 (2)	12 (5)	7 (6)	1 (6)		
D3	13 (8)	14 (8)	10 (7)	6 (7)	14 (6)	7 (9)		
D4	5 (1)	3 (4)	5 (5)	1 (4)	14 (1)	9 (8)		

Table IV. A solution of the example (minimize exposure time).

Bus routes							Exposure time	Buses' routing time	Waiting time
Bus 1	D1 (0)	p1 (2)	d2 (5)	D3 (7)			5	5	2
Bus 2	D1 (0)	p4 (6)	d3 (8)	p3 (13)	d2 (18)	D3 (20)	26	12	0
Bus 3	D2 (0)	p6 (4)	d3 (7)	p2 (15)	d1 (17)	D3 (9)	24	6	0
Bus 4	D2 (0)	p5 (7)	d3 (11)	D3 (13)			11	11	2
Total							66	34	4

and minimizing buses' routing time plus waiting time. Note that buses' routing time does not account for the trip from the last served drop-off to the depot (D^-) because the trip is not subject to any risk exposure. The solutions are tabulated in Tables IV–VII.

The best possible buses' routing time could be obtained in the illustrative example is 25. If one opts to minimize exposure time, then the buses' routing time becomes 34. The best possible of passengers' waiting time at bus stops is 4; both objectives of minimizing exposure time only and waiting time only achieve this result. Minimizing waiting time, however, produces a much worse exposure time, compared with the best exposure time obtained in Table IV. It evidences that minimizing waiting time only may lead to undesired solution with long travel time. It turns out that the single objective of minimizing buses' routing time only, or waiting time only, may yield a solution with much worse exposure time and thus is not suitable for evacuation purpose. Minimizing buses' routing time plus

DOI: 10.1002/atr

Table V. A solution of the example (minimize buses' routing time).

	Bus routes						Exposure time	Buses routing time	Waiting time
Bus 1	D1 (0)	p1 (2)	d2 (5)	p3 (13)	d2 (18)	D3 (20)	23	10	2
Bus 2	D1 (0)	p4 (6)	d3 (8)	p5 (10)	d3 (14)	D3 (26)	22	9	5
Bus 3	D2 (0)	p6 (4)	d3 (7)	p2 (15)	d1 (17)	D3 (19)	24	6	0
Total							69	25	7

Table VI. A solution of the example (minimize waiting time).

	Bus routes						Exposure time	Buses' routing time	Waiting time
Bus 1	D1 (0)	p4 (6)	d2 (16)	D3 (18)			16	13	0
Bus 2	D1 (0)	p1 (2)	d3 (6)	p2 (15)	d3 (20)	D3 (22)	26	13	2
Bus 3	D2 (0)	p6 (4)	d3 (7)	p3 (13)	d2 (18)	D3 (20)	25	11	0
Bus 4	D2 (0)	p5 (7)	d2 (15)	D3 (17)			15	17	2
Total							82	54	4

Table VII. A solution of the example (minimize buses' routing time + waiting time).

	Bus routes						Exposure time	Buses' routing time	Waiting time
Bus 1	D1 (0)	p4 (6)	d3 (8)	D3 (10)			8	5	0
Bus 2	D1 (0)	p1 (2)	d2 (5)	p3 (13)	d2 (18)	D3 (20)	23	10	2
Bus 3	D2 (0)	p6 (4)	d3 (7)	p5 (9)	d3 (13)	D3 (15)	11	8	4
Bus 4	D2 (0)	p2 (15)	d1 (17)	D3 (19)			17	3	0
Total							68	26	6

waiting time obtains a solution with good exposure time and waiting time but inferior to minimizing exposure time. Minimizing exposure time produce the only solution with both best exposure time and best waiting time in the set of evaluated objectives. In summary, the results clearly demonstrate that different objectives of the problems may come up with substantially variant solutions. We conclude that the objective of minimizing exposure time proposed in this paper cannot be obtained by one another objective presented in comparison.

5.2. Real-size example

In this section, we describe computational experiments for a set of randomly generated instances, comprising up to 30 vehicles, 120 pickup requests, and 65 delivery nodes.

A time window $[a_i,b_i]$ is associated with each pickup request node. First, a_i is chosen from a uniform random distribution between $[T_1,T_2]$, where T_1 and T_2 are the minimal and maximal numbers, respectively, and then, b_i is set to a_i+T, indicating that the passengers' instance-specific maximum waiting time is T minutes. In our example, a_i is randomly chosen between [0,100], and T is randomly chosen between [1,50].

Typically, our samples consist of 20–120 pickup requests and 10–65 delivery points. First, we solve a UB (except for scenario M6 in which we just guess a UB) and then apply the Lagrangian relaxation framework to improve the UB. We report the best LB that the Lagrangian method can find within 2 hours and the CPU time taken to solve the instance to optimality if the instance is solved in less than 2 hours. The optimal solutions are marked with asterisks. The UB was compared with the best Lagrangian LB found within 2 hours. The gap is denoted as the difference between the UB and LB divided by the LB. Table VIII shows the results. It is noticed that for those instances the Lagrangian method cannot find an optimal solution within 2 hours, we do not report the number of pickup requests being served because the solutions obtained within 2 hours are infeasible.

Table VIII. Computational results for real-size instances.

Scenario name	Network: No. buses	Network: No. pickup requests	Network: No. delivery nodes	CPU time (seconds): LB	CPU time (seconds): UB	Max. pickups served: LB	Max. pickups served: UB	Solutions: LB	Solutions: UB	Solutions: Gap (%)
M1	20	40	50	>2 hours	<1		40	1510	1516	0.4
M2	25	50	40	188.550	<1	50	50	1530*	1530*	0
M3	16	30	65	>2 hours	<1		30	656.999*	660	0.5
M4	12	45	15	92.121	<1	45	45	1449*	1460	0.8
M5	16	45	15	107.0	<1	45	45	1537*	1537*	0
M6	30	100	20	>2 hours	1.3		100	3105.29	3137	1.0
M7	12	20	10	64.933	<1	20	20	1060*	1065	0.5
M8	25	80	15	>2 hours	<1		80	3636.661	3667	0.8
M9	28	120	30	>2 hours	1.3		120	4629.999	4673	0.9

LB, lower bound; UP, upper bound.
*the optimal solutions are marked with.

Table VIII indicates that the Lagrangian method sometimes completes quickly, but in other times, converges rather slowly. Generally, it appears that the larger the problem size and the tighter the associated time windows, the more likely the Lagrangian method converges slowly. In five out of nine instances, the optimal solutions were not found within 2 hours. The convergence is more relevant to the sub-gradient method, that is, step-size rule of α in Equation (16). In fact, because the step-size rule of α is heuristic, there is a chance that the Lagrangian relaxation method cannot reach to optimal. In contrast, the UB can be solved considerably faster, typically taking less than 1 second to find a good quality solution. With this method, the gap is typically less than 1%. One drawback of the UB is that occasionally, it may not satisfy the pickup request time window when a satisfactory solution is possible. This implies that UB may utilize more buses than necessary, resulting from a myopically search without knowing the information about the next trip. Note that if the given number of buses is not feasible to serve all pickup requests within the associated time windows, then the Lagrangian method may never terminate. However, the UB method can be applied first to secure feasibility.

6. CONCLUDING REMARKS

In this paper, we define and study an optimal bus operating model during an emergency evacuation. A set of constraints or requirements satisfying emergency evacuation needs is carefully addressed. We formulate the problem as a mixed-integer program, which is shown NP-hard.

Lagrangian relaxation was used to develop the solution algorithm. A heuristic is necessary in the Lagrangian framework to obtain a UB. We showed that the resulting Lagrangian relaxation problem is structured as a typical min-cost flow problem on an extended time–space network. Because the min-cost problem possesses integrality, the Lagrangian framework avoids branch-and-bound. Computational experiments were performed in a set of randomly generalized instances of different problem sizes.

The proposed model is for purpose of offline planning, or what-if scenario analysis, to assist no-notice evacuation in an urban area. On a large-scale network, the proposed Lagrangian-based algorithm may not be applicable as constructing the time–space network become memory inhibitive. This is a theoretical feature of the algorithm because of the complexity of the problem that we are solving (i.e., Proposition 1). In such a case, we suggest using the heuristic of UB method to produce a fairly good solution in a quick time. In general, there is always a trade-off between computational performance and solution quality. However, we opt to examine solution quality in depth in this paper, rather than applying simple heuristics in most of studies along this research line.

The deterministic model formulated in the paper is our start. Future research should consider uncertainty and online pickup request information called in real time, rather than a temporally elaborated demand steaming fully known ahead. In a disaster scenario where congestion maybe incurred all around, the assumption of the constant roadway travel time needs to be relaxed. This may require

DOI: 10.1002/atr

obtaining temporally discretized roadway travel time information provided by an external simulation approach. As simulation is not optimization friendly, how to feedback between network performance/travel time information and a routing optimization framework involves significant research effort in further work. In addition, the proposed model is tested with hypothetical instances only because of the lack of data from a real performing agency. Implementation of the proposed method to a real-world case study would be our ongoing research.

LIST OF ABBREVIATIONS AND SYMBOLS

VPR	vehicle routing problem
UB	upper bound
LB	lower bound
VPRTW	shortest path problem with time window
NP	nondeterministic polynomial
LD	a problem specified by Equations (13), (3)-(6), (8)-(12)
LR	a problem specified by Equations (14), (3)-(6), (8)-(12)
H	time horizon
λ_1, λ_2	dual variable
r, θ	dummy nodes
G	graph
G^τ	time expanded graph
τ	time index
$[a_i, b_i]$	time window
α^n	step size
n	iteration index
$P\|r_i\|\Sigma C_i$	parallel machine scheduling with release time, minimizing total job completion time. Following the notation of [21]
D^+	Set of start depots (including pseudo-depots)
D^-	Set of end depots
D	Set of depots, $D = D^+ \cup D^- = \{k \mid k \in D\}$
R	Set of pickup nodes (i.e., pickup requests)
E	Set of delivery nodes (i.e., delivery points).
$C(k)$	Number of buses available at a start depot $k \in D^+$ or capacity of an end depot $k \in D^-$
t_{ij}	Travel time (a positive number) from node $i \in D^+ \cup R \cup E \cup D^-$ to node $j \in D^+ \cup R \cup E \cup D^-$, $t_{ij} > 0$
$T(k)^v$	Time when bus v departs from a depot $k \in D^+$, $T(k)^v \geq 0$
S_i	Service (loading/unloading) time of node $i \in R \cup E$
U_j	Capacity of delivery node $j \in E$, maximal full-bus loads can drop off
M	A constant large number.
x_{ij}^v	1 if bus v is assigned on arc (i,j), 0 otherwise
$T(i)^v$	Time when bus v visits the pickup node $i \in R$
$T(j)_i^v$	Time when bus v visits the delivery node $j \in E$ after visiting $i \in R$.

REFERENCES

1. Chiu Y-C, Zheng H. Real-time mobilization decisions for multi-priority emergency response resources and evacuation groups: model formulation and solution. *Transportation Research* 2007; **43E**(6):710–736.
2. Chiu Y-C, Zheng H, Villalobos J, Gautam B. Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Transactions* 2007; **39**(1) 83–94.
3. Chiu Y-C, Zheng H, Villalobos JA, Peacock W, Henk R. 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* 2008; **5**(1) Article 34.
4. Zheng H, Chiu Y-C, Mirchandani PB, Hickman M. Modeling of evacuation and background traffic for optimal zone-based vehicle evacuation strategy. *Transportation Research Record* 2009; **2196**: 65–74.

DOI: 10.1002/atr

5. Perkins JA, Dabipi IK, Han LD. *Modeling Transit Issues Unique to Hurricane Evacuations: North Carolina's Small Urban and Rural Areas*; Transportation Institute, North Carolina Agricultural and Technical State University, Urban Transit Institue, Transportation Institute, 2001Greensboro, NC.
6. Transportation Research Board. 2008. The Role of Transit in Emergency Evaucation, TRB Special Report 294. Transportation Research Board, Washington, D.C.
7. Zimmerman C, Brodesky R, Karp J. *Using Highways for No-Notice Evacuations: Routes to Effective Evacuation Planning Primer Series*. USDOT: Washington, D.C. 2007.
8. Abdelgawad H, Abdulhai B, Wahba M.. 2010; Optimizing mass transit utilization in emergency evacuation of congested urban areas *Proceedings of the 89th Annual Meeting of the Transportation Research Board*, Washington,DC.
9. Sayyady F, Eksioglu SD. Optimizing the use of public transit system during no-notice evacuation of urban areas. *Computers and Industrial Engineering* 2010; **59**(4):488–495.
10. Zografos KG, Androutsopoulos KN. 2008; A decision support system for integrated hazardous materials routing and emergency response decisions. *Transportation Research* **16C**(6):684–703.
11. Pages L, Jayakrishnan R, Cortes CE. Real-time mass passenger transport network optimization problems. *Transportation Research Record* 2006; **1964**:229–237.
12. Sbayti H, Mahmassani HS. Optimal scheduling of evacuation operations. *Transportation Research Record* 2006; **1964**:238–246.
13. He S, Zhang L, Song R, Wen Y, Wu D. 2009. Optimal transit routing problem for emergency evacuations *Proceedings of 88th Annual Meeting of Transportation Research Board*, Washington, DC.
14. Chen C-C, Chou C-S. 2009; Modeling and performance assessment of a transit-based evacuation plan within a contraflow simulation environment. *Transportation Research Record* **2091** 40–50.
15. Elmitiny N, Ramasamy S, Radwan E. Emergency evacuation planning and preparedness of transit facilities: traffic simulation modeling. *Transportation Research Record* 2007; **1992**121–126.
16. Haghani A, Hu H, Tian Q. 2003; An optimal model for real-time emergency vehicle dispatching and routing. *Proceedings of 83rd Annual Meeting of Transportation Research Board*, Washington, DC.
17. Mastrogiannidou C, Boile M, Golias M, Theofanis S, Ziliaskopoulos A. 2009. Using transit to evacuate facilities in urban areas: a micro-simulation based integrated tool *Proceedings of 88th Annual Meeting of Transportation Research Board*, Washington, DC.
18. Desrochers M., Desrosiers J, Solomon M. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 1992; **40**(2) 342–354.
19. Kohl N, Madsen OBG. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research* 1997; **45**(3):395–406.
20. Dror M. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* 1994; **42**(5) 977–978.
21. Graham RL, Lawler EL, Lenstra JK, Kan AHGR. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* 1979; **5** 287–326.
22. Lenstra JK, Kan AHGR, Brucker P. Complexity of machine scheduling problems. *Annals of Discrete Mathematics* 1977; **1**:343–362.
23. Held M, Karp RM. The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming* 1971; **1**(1):6–25.

DOI: 10.1002/atr