A Two-step Polynomial-time Approach for Bus Evacuation Planning

Yuanyuan Feng,Yi Cao, and Shuanghua Yang

***Abstract*—In large-scale disasters, as an important part of the rescue management, buses, due to the large capacity and manageability, are often used to evacuate car-less people. Timing is always the main concern in any evacuation planning. Hence, not only is minimizing evacuation time the primary goal of a bus evacuation problem (BEP), but computation time to solve a BEP is also crucial. Nevertheless, minimizing bus evacuation time is NP-hard due to the combinatorial nature of the integer linear programming problem associated. This makes optimization of the BEP for large-scale disasters intractable. Practically, sub-optimal but polynomial-time algorithms for bus evacuation planning are desired. In this work a two-step polynomial-time approach, called Network Flow Planning (NFP) approach is proposed. In the NFP, firstly, an aggregated network flow model, which minimizes the total travel time of all the buses, is adopted. It is proven that the model can be solved as a linear programming problem in a polynomial time. The minimum total travel time solution is then converted to approximate the minimum evacuation time solution by assigning evacuation tasks to all buses as equally as possible in the second step. The proposed post-processing algorithm of the network flow solution consists of task construction and task assignment algorithms. To verify the effectiveness of the NFP approach, several numerical case studies are presented in this paper. In a Monte Caro simulation study, randomized cases are used to demonstrate the superiority of the proposed approach over CPLEX, a genetic algorithm and an approximation algorithm based solutions. Furthermore, a real-world large-scale flood disaster case in Xingguo, China is also studied. The second case illustrates the efficiency and practical value of the proposed approach in large-scale evacuations.**

***Index Terms*—bus evacuation problem, network flow model, polynomial time, large-scale disaster**

I. Introduction

DISASTERS, either man-made (such as terrorist attacks and chemical explosions) or natural (such as earthquakes, floods, and hurricanes), could result in devastating losses of lives and properties [1]. Evacuation from disaster regions is one of the most effective operations to protect people from the possible impact of disasters and is an important part of disaster management. Among various evacuation vehicles, privately-owned vehicles are difficult to coordinate in an emergency and potentially lead to exacerbating traffic congestion and hindering the overall evacuation process. Moreover, due to their unstable physical and mental condition in disasters, evacuees may be unable to drive their own cars. Therefore, high-capacity public vehicles should be provided as the mode of evacuation [2] to substantially improve the evacuation process and reduce evacuation casualties.

Time is always crucial in response to any disaster [3]. In the response phase, any kind of delay may cause some preventable losses. The longer people are exposed to a disaster, the more likely they are to suffer casualties. Among many efforts to minimize evacuation time, Krasko and Rebennack [4] formulated a two-stage model focusing on minimizing the expected damages of the post-fire debris flow hazard, which is not applicable to other disasters. Pereira and Bish [5] drew up the evacuation plan with the objective of minimizing the total exposure time while evacuees were assumed to arrive pickup points at constant arrival rates. And some researchers [6, 7, 8, 9] aimed to maximize the number of evacuated victims in a restricted time. Notwithstanding, they all assumed that not all evacuees may be transported, thus some of them may be abandoned due to their remote positions, which is unfair and inhumane in disasters. In addition, minimizing total operation time or cost [2, 10, 11, 12, 13] are other common objectives. But with the total travel time minimum, some evacuees may be detoured or sent to a remote shelter, and the final evacuation completed time may be much longer than the average travel time of all the buses. Besides, some authors also considered other factors *e.g.* commodity distribution [12, 14], psychological cost [15], time windows [16] or even multi-objectives [15, 17] in their evacuation models. No doubt, among all these objectives, transporting all the evacuees to safe shelters with limited fleets of buses in minimum time is the top priority in any evacuation [18]. Thus, bus evacuation plans should be formulated to minimize the evacuation time.

Although the optimality of an evacuation plan is important, in a real disaster situation, the evacuation agent should make an executable plan as quick as possible, at least, much quicker than the evacuation time itself. Unfortunately, optimization of bus evacuation planning is NP-hard [19, 20]. In the literature, most algorithms for bus evacuation problems (BEP) tend to search and iterate among all possible paths to meet the constraint conditions in either deterministic or heuristic ways.

In deterministic aspect, branch-and-bound framework [21], branch-and-price [22] and cutting plane scheme [23] have been developed for BEPs and are more efficient than brutal-force approaches. But they are not able to change the NP hard nature of the problem, hence cannot solve the problem in a reasonable time for large-scale disasters. On the other side, approximation algorithms for sub-optimal solutions are also sought. Pedrosa and Schouery [20] presented a 10.2-approximation algorithm for cases where all buses must start from one same depot and a 4.2-approximation algorithm for the further simplified

Yuanyuan Feng is with the College of Computer Science and Technology, Zhejiang University,866 Yuhangtang Rd, Hangzhou, China, e-mail: fengyy@zju.edu.cn .

Yi Cao and Shuanghua Yang are with the College of Chemical and Biological Engineering, Zhejiang University.

cases in which the shelter capacity is unlimited. In addition to their unpractical assumptions of the BEP problem, such high approximation ratios make the evacuation plans obtained inefficient in practical situations.

In order to overcome the scale limitation of deterministic algorithms[24], many researchers developed heuristic algorithms for BEPs. Goerigk et al. [25] developed a scenario-generation algorithm which iteratively adds new scenarios to the sub-problem that is being solved. Oh et al. [6] proposed a cooperative multiple agent-based algorithm to efficiently find a sub-optimal solution for a fleet of aerial vehicles. Moreover, genetic algorithms [8, 9, 26, 27], simulated annealing algorithm [2, 11], particle swarm optimization algorithm [7] and neighborhood search approach [28] have also been applied to solve BEPs. Although most heuristic algorithms are designed to be able to give a feasible solution at any time, they normally take many iterations to obtain an acceptable approximate solution, while repetitive iterations and constraint check are time consuming. For example, the largest problem shown by Dikas and Minis [28] with 200 pickup points, 10 shelters and 20 vehicles needed about 10 hours CPU time to converge to a solution, which is too long to be acceptable in an emergency situation. Compared to the evacuation time, the time that is consumed to make up an evacuation plan should be negligible. Nevertheless, for large-scale disasters, getting an optimal or acceptable sub-optimal evacuation plan in a polynomial time is still open and challenging.

This work proposed a two-step polynomial-time approach, referred to as Network Flow Planning (NFP), for large-scale BEPs. Firstly, a network flow model with the objective of minimizing the total travel time is formulated by aggregating the 0-1 integer programming BEP model over all buses and trips. In the network flow model, it is assumed that the number of evacuees in a pick-up point is multiple to the bus capacity. This assumption not only simplifies the mathematical problem, but also makes the rescue plan practically executable for large-scale disasters. It is proven that aggregated network flow model is a linear programming problem, hence, can be solved a polynomial time. A lower bound of the minimum evacuation time is then obtained by dividing the minimum total travel time by the number of available buses. In other words, assigning the bus flows, the solution of the network flow model, evenly into each bus achieves the minimum evacuation time. Therefore, in the second step, a post-processing algorithm consisting of task construction and task assignment sub-algorithms is developed to convert the solution of the network flow model into a bus evacuation plan. The post-processing algorithm gives consideration to both the continuity of the tasks assigned to a single bus and the evenness of travel time distributed to all buses, such that a bus evacuation plan can be derived in polynomial time and the actual evacuation time of the plan approximates the minimum evacuation time tightly. Compared with the lower bound of the minimum evacuation time as a reference to the optimal solution and other algorithms available in the literature, random cases are studied to validate the efficiency of the NFP approach. Finally, the new approach is applied to a real-world large-scale disaster, Xingguo flood, occurred in China in 2019. It is shown that this approach proposed can provide an appropriate evacuation plan in a polynomial time.

The main contributions of this paper are as follows:

C1. A two-step polynomial-time approach, NFP, is proposed for large-scale BEPs with the objective of minimizing the evacuation time.

C2. A network flow model with the objective of minimizing the total travel time for BEPs is formulated and it is proven that it can be solved by linear programming in a polynomial time.

C3. A mathematical theorem to evenly pair two groups of numbers is proposed and proven. The theorem provides a theoretical basis to develop the post-processing algorithm.

C4. For post-processing, a task construction algorithm is developed to transfer the solution of the network flow model into evacuation tasks based on the evenly pairing theorem.

C5. Based on the same theorem, a task assignment algorithm is designed to allocate the constructed tasks to all buses as even as possible.

C6. A Monte Carlo simulation study confirms the superiority of the NFP approach proposed over genetic algorithm(GA), approximation algorithm and the commercial general integer programming solver, CPLEX.

C7. A real-world large-scale case study, Xingguo flood, occurred in China in 2019 is successfully conducted using the NFP approach.

The rest of this paper is organized as follow. In Section II, the BEP for minimizing evacuation time and the aggregated network flow model for minimizing the total travel time are presented, whilst, the post-processing algorithm is proposed in Section III. Then the proposed NFP approach is applied to an illustrative example, some randomized cases and a real disaster scenario, Xingguo flood disaster occurred in China in 2019, in Section IV. Finally, the work is summarized with some concluding remarks together with a discussion of future work in Section V.

II. BEP AND NETWORK FLOW MODELS

A. A formulation of BEP with bus indices

The BEP was originally proposed by Bish [29]. A simplified version of the problem presented below is to find a schedule for a set of buses to transport all evacuees from the pickup points to the capacity restricted shelters so that the evacuation time is minimized. For large-scale disasters, the number of evacuees is usually much larger than the capacity of a bus, and each pickup pint always needs to be visited by several buses several times. Thus after visiting a pickup point, the rescue buses often need to directly head to shelters with full loads of evacuees. To simplify the problem and make evacuation plans practically executable, the number of evacuees on each pickup point is rounded up to multiple times of the bus capacity. Other assumptions are listed as follows:

A1. The number and geographical locations of available buses are given when a disaster occurs.

A2. The capacities of buses are identical.

A3. Buses are assumed being operated continuously, and the loading/unloading time at pickup points/shelters and other lost times are ignored, for simplicity.
A4. The travel of a bus from a depot or shelter, passing by a pickup point, to a shelter is defined as a trip. Buses depart from depots on the first trip and stay at shelters after finishing the last trip.
A5. The numbers, locations and the capacities of shelters are known.
A6. As the capacities of shelters are much larger than that of a bus in a large evacuation scenario, the number of evacuees that a shelter can served are lumped into multiple times of a bus capacity.
A7. The travel time between nodes (depots, pickup points, shelters) is constant and given.
A8. Inter-movements between pickup points and between shelters are forbidden.
A9. Each route, between a pickup point and a shelter can be taken by several buses.

Based on the above assumptions, an exact bus evacuation model can be formulated. Firstly, symbols used in the model are defined as follows.

Sets:

D	Index set of depots where all the buses originate from, $D = \{1, \ldots, n_d\}$.
P	Index set of pickup points where evacuees are gathered and waiting for buses, $P = \{1, \ldots, n_p\}$.
S	Index set of shelters which are the destinations of evacuees and used buses, $S = \{1, \ldots, n_s\}$.
R	Index set of trips each bus might possibly take, $R = \{1, \ldots, n_r\}$.
B	Index set of all the available buses, $B = \{1, \ldots, n_b\}$, and a bus $b \in B$ sets off from a depot $k_b \in D$.
I_{AB}	Route set from A to B, $I_{AB} = \{(i,j) \mid i \in A, j \in B\}$, $A = D, P, S$, $B = P, S$.
I	Permissible route set, $I = I_{DP} \cup I_{PS} \cup I_{SP}$.

Parameters:

B_k	Number of buses available at depot $k \in D$.
δ_{ij}	Travel time (a positive number) from node i to node j, where $(i,j) \in I$.
Q_i	The number of pickup requests on pickup point $i \in P$.
U_j	The capacity of shelter $j \in S$.

Decision variables:

x_{ij}^{br}	a binary variable, represents whether bus $b \in B$ is assigned to go from node i to node j, $(i,j) \in I$ on trip r.

With symbols defined above, the binary BEP model is as follows.

$$\min \max_{b\in B} \sum_{r\in R} \sum_{(i,j)\in I} \delta_{ij} x_{ij}^{br} \tag{1}$$

$$\text{s.t.} \quad \sum_{b\in B} \sum_{i\in P} x_{ki}^{b1} \le B_k, \quad \forall k \in D \tag{2}$$

$$\sum_{r\in R\setminus 1} \sum_{b\in B} \sum_{(k,i)\in I_{DP}} x_{ki}^{br} = 0 \tag{3}$$

$$x_{k_b i}^{b1} = \sum_{j\in S} x_{ij}^{b1}, \quad \forall b \in B \tag{4}$$

$$\sum_{l\cup S} x_{li}^{br} = \sum_{m\in S} x_{im}^{br}, \quad \forall i \in P, b \in B, r \in R \setminus 1 \tag{5}$$

$$\sum_{l\in P} x_{lj}^{br} \ge \sum_{m\in P} x_{jm}^{b,r+1}, \forall b \in B, j \in S, r \in R \setminus n_r \tag{6}$$

$$\sum_{(i,j)\in I_{PS}} x_{ij}^{br} \le 1, \quad \forall b \in B, r \in R \tag{7}$$

$$Q_i \le \sum_{j\in S} \sum_{b\in B} \sum_{r\in R} x_{ij}^{br}, \quad \forall i \in P \tag{8}$$

$$U_j \ge \sum_{i\in P} \sum_{b\in B} \sum_{r\in R} x_{ij}^{br}, \quad \forall j \in S \tag{9}$$

$$x_{ij}^{br} \in \{0,1\}, \quad \forall (i,j) \in I, b \in B, r \in R \tag{10}$$

The objective function (1) is to minimize the overall evacuation time, i.e., the evacuation time until the last evacuee is brought to a shelter. Constraints (2) and (3) ensure that all the buses can only travel out of depots in their first trip and the numbers of them are not larger than that parked in the depots. Constraints (4)-(6) ensure the continuity of bus travel *i.e.* buses involved in evacuation can only set off from D at the first trip and park at shelters, and the rescue operation of each bus does not pause in the middle. Constraint (7) ensures that in a single trip a bus can only serve one pickup point and send those evacuees to one shelter at most. Constraints (8) and (9) ensure that all evacuees are picked up, and each shelter can accommodate evacuees no more than its capacity.

Since x_{ij}^{br} are integer variables, the combinatorial nature makes the model NP-hard. The model has $N = (n_d n_p + (2n_r - 1)n_p n_s) n_b n_r \ge (n_d n_p + (2n_r - 1)n_p n_s)Q$ binary variables, where $Q = \sum_{i\in P} Q_i$, correspondingly, 2^N possibilities, which grows exponentially as N increases. For instance, in Xingguo flood disaster, occurred in China in 2019, there are 25 shelters and 22 pickup points with about 66,000 evacuees. If the bus capacity is 50, then $N \ge 726,000$, and there are more than 10^{218547} possibilities. Even if each possibility can be evaluated in a microsecond, to evaluate all possibilities will require more than 10^{5203} years. Therefore, it is impossible to solve such a large-scale BEP in a reasonable time.

Nevertheless, due to the uniformity of buses, bus trips are exchangeable among buses which arrive at same point at same time. This feature can be used to simplify the model in order to get an acceptable solution in a short time.

B. A network flow model

The binary decision variables of the BEP model with bus indices can be aggregated over all buses and trips to simplify the model and eliminate the symmetry, *i.e.*

$$X_{ij} = \sum_{b\in B}\sum_{r\in R} x_{ij}^{br},\ \forall (i,j) \in I \tag{11}$$

Since the aggregation merges all the trips together, the sequential order of bus trips is not determined in the new network flow model. Thus, the objective function has to be changed to minimize the total travel time. The new network flow model is as follows:

$$\min \quad \sum_{i\in D\cup P\cup S}\sum_{j\in P\cup S} \delta_{ij}X_{ij} \tag{12}$$

$$\text{s.t.} \quad \sum_{(k,i)\in I_{DP}} X_{ki} \geq \min\left(\sum_{k\in D} B_k, \sum_{i\in P} Q_i\right) \tag{13}$$

$$\sum_{i\in P} X_{ki} \leq B_k, \quad \forall k \in D \tag{14}$$

$$\sum_{l\in D\cup S} X_{li} - \sum_{m\in S} X_{im} = 0, \quad \forall i \in P \tag{15}$$

$$\sum_{m\in P} X_{jm} - \sum_{l\in P} X_{lj} \leq 0, \quad \forall j \in S \tag{16}$$

$$-\sum_{j\in S} X_{ij} \leq Q_i, \quad \forall i \in P \tag{17}$$

$$\sum_{i\in P} X_{ij} \leq U_j, \quad \forall j \in S \tag{18}$$

$$X_{ij} \in \mathbb{R}_+ \tag{19}$$

The objective function (12) is to minimize the total travel time of all buses. Constraint (13) is added to make sure all available buses have been efficiently used. Constraints (14)-(18) are corresponding to the constraints (2), (5), (6), (8), (9). And other constraints in the exact model are omitted as they are no longer relevant after the trip aggregation.

After aggregating, decision variables and constraints decrease to around $N_v = n_d n_p + 2n_p n_s$ and $N_c = n_d + 2(n_p + n_s) + 1$ respectively. More importantly, although the aggregated model is still an integer linear programming, the solutions of the linear programming is integer hence the problem is solvable in polynomial time as shown in the following theorem.

Theorem 1. *The integer programming problem in (12) - (19) is a linear programming problem solvable in a polynomial time.*

Proof. According to Ghouila-Houri [30], matrix $A = (a_{ij}) \in \mathbb{Z}^{m\times n}$ is unimodular iff there is a row partition $A = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix}^T$ such that:

$$\sum_{i\in A_1} a_{ij} - \sum_{i\in A_2} a_{ij} \in \{-1, 0, 1\}, \quad \forall j = 1, 2, \ldots, n \tag{20}$$

The partition with all constraints as A_1 and A_2 as $\emptyset$ satisfies the unimodular condition (20). Then, according to Hoffman and Kruskal [31], the polyhedron defined by such constraints $\{x|Ax \leq b, x \geq 0\}$, for integer A and b, has the integral property. In other words, solution of the linear programming problem is integer. According to [32], the complexity of this kind of combinatorial linear programs is $O(m^4 + m^3 + mT(A))$, where m is the number of rows of A and $T(A)$ is a polynomial in the size of A. □

Solving the network flow model gives bus flows as X_{ij}, $(i,j) \in I$ and the minimized total travel time, T_t^*, for post-processing to construct an evacuation plan approximating the minimum evacuation time, T_e^*, described in the next section.

III. Post-processing Algorithm of the Network Flow Model

Although the network flow model can be solved in a polynomial time, the result obtained is the minimum total travel time, T_t^*, instead of the minimum evacuation time, T_e^*. Moreover, the solution only indicates the number of buses that pass through a road during the whole evacuation process without a detailed evacuation plan. In order to get a workable evacuation plan, a second step to post-process the network flow solution is necessary, which in turn will significantly impact actual evacuation time. Intuitively, if the minimum total travel time can be equally allocated to all buses used, then it minimizes the evacuation time. This observation not only gives a lower bound of the minimum evacuation time, *i.e.* $\underline{T_e^*} = T_t^*/n_b$, but also provides a guideline to design a post-processing algorithm to allocate evacuation tasks to available buses as equally as possible. The network flow model and the post-processing algorithm together construct a two-step polynomial-time approach, so called Network Flow Planning approach (NFP). The structure of NFP is shown in Figure 1, and in this section the NFP approach is proposed to transform the solution of the network flow model into a desired bus evacuation plan.

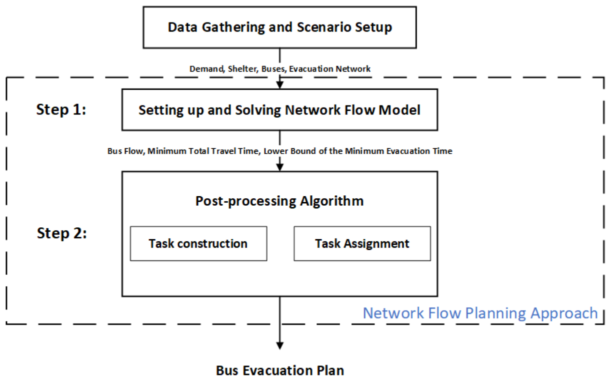

Data Gathering and Scenario Setup
Demand, Shelter, Buses, Evacuation Network
Step 1:
Setting up and Solving Network Flow Model
Bus Flow, Minimum Total Travel Time, Lower Bound of the Minimum Evacuation Time
Step 2:
Post-processing Algorithm
Task construction
Task Assignment
Network Flow Planning Approach
Bus Evacuation Plan

Fig. 1. The structure of the network flow planning approach for bus evacuation planning

At a pickup point, $i \in P$, there are Q_i in and out bus flows determined by the solution obtained. These in- and out- flows can be paired into Q_i tasks. The l^{th} quadruple task, $\mathcal{T}_l = \{d_l, p_l, s_l, t_l\}$, indicates that a bus starts from a shelter or a depot $d_l \in S \cup D$ to pick up evacuees in node $p_l \in P$ and send them to a shelter $s_l \in S$ with total task time, t_l. Due to constraints (15) and (16), if $d_l \in S$, there is at least a task $\mathcal{T}_m = \{d_m, p_m, s_m, t_m\}$ ending at the same shelter, *i.e.* $d_l = s_m$ can be followed by task $\mathcal{T}_l$.

Then several such tasks connected continuously will construct a rescue plan of bus $b \in B$, *i.e.* $\mathcal{P}_b = \{\mathcal{T}_1^b, \mathcal{T}_2^b, \dots, \mathcal{T}_r^b\}$, where $\mathcal{T}_r^b = \{d_r^b, p_r^b, s_r^b, t_r^b\}$ denotes task $\mathcal{T}_r = \{d_r, p_r, s_r, t_r\}$ is the r^{th} task assigned to bus b. Therefore, the post-processing of network flow solution consists of two stages, constructing rescue task list, $\mathcal{T} = \{\mathcal{T}_l | l = 1, \dots, Q\}$, by pairing in- and out- flows at the same pickup points firstly, then assigning all the tasks to all buses to form a complete rescue plan, $\mathcal{P} = \{\mathcal{P}_b | b \in B\}$. If all in- and out- flows obtained could be paired into appropriate tasks and these tasks could be assigned as even as possible to available buses, the solution of network flow model is then converted into a bus evacuation plan, $\mathcal{P}$ with the actual evacuation time, T_e, tightly approximating T_e^*, where

$$T_e = \max_{b \in B} \sum_{r=1}^{Q} t_r^b \tag{21}$$

A. Task construction

At a pickup point, $i \in P$, there are Q_i in- and out- flows, hence there are total $Q_i!$ possibilities to pair them. The way to pair tasks will affect the evacuation efficiency. An efficient algorithm to convert the network flow solution into appropriate rescue tasks is proposed as follows.

To achieve the final goal of post-processing that the travel time of buses is allocated as even as possible, tasks are constructed as even as possible so that the travel time of the largest task is minimized as well. To achieve this, firstly, at each demand point i, inflows and outflows are sorted against their travel time. Then sorted inflows and outflows are paired reversely, namely matching the minimum time inflow with the maximum time outflow.

Theorem 2. *Assume inflow travel times $t_1^I \le t_2^I \le \dots \le t_m^I$ and outflow travel times $t_1^O \ge t_2^O \ge \dots \ge t_m^O$. Then pairing (t_i^I, t_i^O), $1 \le i \le m$ minimizes the longest pair travel time among all possible pairing.*

Proof. Let $t_i = t_i^I + t_i^O$, $i = 1, \dots, m$ and $t_k = \max_{1 \le i \le m} t_i$. If there is another pairing method, $t_i' = t_i'^I + t_i'^O$, $i = 1, \dots, m$ and $t_k > \max_{1 \le i \le m} t_i'$, than $t_k \notin \{t_1', \dots, t_m'\}$, *i.e.* $(t_k^I, t_k^O) \notin \{(t_i'^I, t_i'^O) | i = 1, \dots, m\}$. This means t_k^I is paired with anther outflow $t_p^O < t_k^O$, $p > k$ (descending) and t_k^O is paired with another inflow $t_q^I < t_k^I$, $q < k$ (ascending). Since $t_p^I > t_k^I$ and $t_q^O > t_k^O$, t_p^I and t_q^O have to be paired with t_j^O, $j > k$ and t_i^I, $i < k$ respectively. Continuing this process, there are at least one inflow and one outflow, which are larger than t_k^I and t_k^O respectively, left and made a pair. Therefore, in this pairing method, the largest travel time can not be smaller than $t_k^I + t_k^O$. □

To conclude, for given inflows and outflows of a pickup point, pairing them reversely is an effective way to guarantee the task travel time distributed as even as possible and the maximum task travel time minimized. Algorithm 1 describes this procedure in more detail.

Algorithm 1 Reverse pairing of task construction

```
Input: An case of BEP ;
    $NetworkFlowModel$ and its integral solution and objective, $X_{ij}$;
Output: the list of all the tasks, $\mathcal{T}$;
 1: function ReversePair($X_{ij}$)
 2:     $\mathcal{T} \leftarrow \emptyset$;
 3:     for $i \in P$ do :
 4:         $\{(d_l, p_l, t_l^I) | l = 1, \dots, Q_i\} \leftarrow \{X_{ji} | \forall j \in D \cup S\}$ times trip $(d_l, p_l), d_l = j, p_l = i$;
 5:         $\{(p_l, s_l, t_l^O) | l = 1, \dots, Q_i\} \leftarrow \{X_{ij} | \forall j \in S\}$ times trip $(p_l, s_l), p_l = i, s_l = j$;
 6:         sort $(d_l, p_l, t_l^I)$ ascending based on $t_l^I, l = 1, \dots, Q_i$
 7:         sort $(p_l, s_l, t_l^O)$ descending based on $t_l^O, l = 1, \dots, Q_i$
 8:         $\mathcal{T}_l \leftarrow \{(d_l, p_l, s_l, t_l) | t_l = t_l^I + t_l^O\}$
 9:         $\mathcal{T} \leftarrow \{\mathcal{T}; \mathcal{T}_l\}$
10:     end for
11:     sort $\mathcal{T}$ descending based on $t_l, l = 1, \dots, Q$
         return $\mathcal{T}$
12: end function
```

A tighter lower bound of the evacuation time can be determined by selecting the larger one between the average travel time and the maximum task time, *i.e.*

$$\underline{T_e} = \max\left\{\underline{T_e^*}, \max_{1 \le i \le Q} t_i\right\} = \max\left\{\frac{T_t^*}{n_b}, \max_{1 \le i \le Q} t_i\right\} \tag{22}$$

where $\underline{T_e^*}$ is a lower bound of the minimum evacuation time, which equals to the minimum total travel time of all the buses, T_t^* divided by the number of buses, n_b, *i.e.*, $\underline{T_e^*} = \frac{T_t^*}{n_b} \le T_e^*$, and t_i is the travel time of the constructed task $\mathcal{T}_i$. Since the network flow model is to minimize the total travel time, it may result in a solution which has a very large single task time. In this situation, evenly distributing other tasks is not able to approximate the minimum evacuation time well if such a large task is avoidable. In order to check whether a large task time is avoidable, the network flow model is modified by adding an extra constraint to disable the particular route, then solved accordingly. The new solution will be reversely paired and a new lower bound will be calculated. If the new lower bound is less than previous one, then the previous solution will be replaced by the new one for further processing. Algorithm 2 describes the check process in detail.

Algorithm 2 Check of the maximum task

```
Input: An case of BEP ;
        $NetworkFlowModel$ and its integral solution and objective, $X_{ij}$, $T_t^*$;
        the list of all the tasks attained from Algorithm 1, $\mathcal{T}$;
Output: the updated list of all the tasks, $\mathcal{T}$;
        Lower bound of evacuation time, $\underline{T_e}$.
1: function TASKCHECK($\mathcal{T}$, $NetworkFlowModel$, $T_t^*$)
2:     $E_t \leftarrow mean\{\mathcal{T}.t_i | i = 1, \dots, Q\}$
3:     $\mathcal{D}_t \leftarrow variance\{\mathcal{T}.t_i | i = 1, \dots, Q\}$
4:     $pOutlier \leftarrow 2$
5:     while $\mathcal{T}.t_1 > E_t + pOutlier * \mathcal{D}_t$ do
6:         Forbid the larger path in $\mathcal{T}_1$ of $NetworkFlowModel$
7:         $\hat{X}_{ji} \leftarrow$Solving $NetworkFlowModel$
8:         $\mathcal{T}_{new} \leftarrow$ REVERSEPAIR($\hat{X}_{ji}$)
9:         if $\mathcal{T}_{new}.t_1 < \mathcal{T}.t_1$ then
10:            $\mathcal{T} = \mathcal{T}_{new}$
11:        else
12:            break
13:        end if
14:    end while
15:    $\underline{T_e} \leftarrow \max\left(\frac{T_t^*}{n_b}, \mathcal{T}.t_1\right)$
16: return $\mathcal{T}$, $\underline{T_e}$
17: end function
```

Algorithm 3 Task construction with minimum ϕ tasks

```
Input: The task list $\mathcal{T}$ and a lower bound of evacuation time $\underline{T_e}$ obtained from Algorithm 2;
        The number of minimum task, $\phi$
Output: The list of all the tasks, $\mathcal{T}$;
1: $(d_l, p_l, t_l^I)$ , $(p_l, s_l, t_l^O) \leftarrow$ splitting $\mathcal{T}$
2: Sort $(d_l, p_l, t_l^I)$ and $(p_l, s_l, t_l^O)$ of each demand point $p_l \in P$ ascending based on $t_l^I$ and $t_l^O$ respectively
3: $\mathcal{T}^f \leftarrow$ pairing $\{(d_l, p_l, s_l, t_l) | t_l = t_l^I + t_l^O\}$
4: Sort $\mathcal{T}^f$ ascending based on $t_l$
5: $\mathcal{T}^r \leftarrow \left\{\mathcal{T}_l^f | l = \phi + 1, \dots, Q\right\}$
6: $\mathcal{T}^r \leftarrow$ splitting $\mathcal{T}^r$ and reverse pairing them
7: $\mathbf{p_l} \leftarrow \arg_l(\mathcal{T}^r.t_l > \underline{T_e})$
8: if $\mathbf{p_l} \neq \emptyset$ then
9:     for $i \in \mathbf{p_l}$ do
10:        Sort $(i, s_l, t_l^O)$ of demand point $i$ descending based on $t_l^O$
11:        Replace $(i, s_l, t_l^O)$ of $\{\mathcal{T}^f | p_l = i\}$
12:    end for
13:    Sort $\mathcal{T}^f$ ascending based on $t_l$
14:    $\mathcal{T}^r \leftarrow \left\{\mathcal{T}_l^f | l = \phi + 1, \dots, Q\right\}$
15:    $\mathcal{T}^r \leftarrow$ splitting $\mathcal{T}^r$ and reverse pairing them
16: end if
17: $\mathcal{T}^s \leftarrow \left\{\mathcal{T}_l^f | l = 1, \dots, \phi\right\}$
18: $\mathcal{T}^s \leftarrow$ splitting $\mathcal{T}^s$ and reverse pairing them
19: $\mathcal{T} \leftarrow \{\mathcal{T}^s; \mathcal{T}^r\}$
20: sort $\mathcal{T}$ descending based on $t_l, l = 1, \dots, Q$
21: return $\mathcal{T}$
```

After the reverse pairing and the check of the maximum task, all the tasks are constructed with travel times as even as possible. However, the number of tasks may not be an integer times of the number of buses so that the numbers of tasks assigned to individual buses are uneven. In this case, it is desired that the extra tasks some buses have to take remain as little travel time as possible. Therefore, Q in- and out- flows are divided into two groups, ϕ minimum in- and out- flows and rest $Q - \phi$ flow pairs. ϕ is defined to be the remainder of Q/n_b. These two groups of flows are to be paired reversely to pledge the evenness. No doubt, this kind of grouping may increase the maximum travel time of all the tasks. To avoid the increase of lower bound caused by the selection of minimum ϕ tasks, the lower bound is checked and the tasks will be updated if there are travel time of tasks exceeding the lower bound derived by Algorithm 2. The more detail of constructing tasks with minimum ϕ tasks is described in Algorithm 3.

B. Task assignment

After the task construction, all tasks end at shelters. Each bus takes an initial task from a depot to a shelter. The remaining tasks all start from shelters. Therefore, the aim of assignment is to assign tasks from shelters to shelters to most appropriate buses such that the travel times among buses are as even as possible.

In order to keep the actual evacuation time T_e as low as possible, the target evacuation time, T_r, is introduced and monitored along the process of task assignment. Initially, $T_r = \max\{\alpha \underline{T_e^*}, \max_{1 \leq i \leq Q} t_i\}$, where $\underline{T_e^*}$ is the lower bound of the minimum evacuation time, inherited from the network flow model, and $\alpha > 1$ is introduced to control the actual evacuation time, which is a tuning parameter of the algorithm. Furthermore, some notations are introduced to facilitate task assignment. Firstly, the set of buses parked at shelter i is denoted as B_i. Assuming a partially assigned plan for bus b with m tasks assigned, $\mathcal{P}_b = \{\mathcal{T}_1^b, \dots, \mathcal{T}_m^b\}$, then the remaining capacity of bus b is defined as $C_b = T_r - \sum_{1 \leq k \leq m} t_k^b$. Finally, let $\mathcal{T}_l, 1 \leq l \leq Q$ without superscript denotes the unassigned task. With these notations, assigning task $\mathcal{T}_l$ to bus b means not only C_b will decrease by t_l but also bus b will end at shelter s_l , *i.e.* $B_{s_l}^{new} = B_{s_l}^{old} \cup b$, hence can undertake the tasks starting from shelter s_l. The total travel time of unassigned tasks starting from shelter i can be calculated as $\sum_{d_l = i} t_l$ and total available bus remaining capacity is $\sum_{b \in B_i} C_b$. Then, the total shortage of bus capacity of a shelter can be determined as $\overline{T}_i = \sum_{d_l = i} t_l - \sum_{b \in B_i} C_b$.

Essentially, task assignment is to pair the assigned tasks of buses with the unassigned tasks. Therefore, Theorem 2 is still applicable to task assignment to achieve an evenly paired result and minimize the maximum travel time of buses. To apply Theorem 2, the assigned task time of a bus is reversely represented by C_b, whilst the unassigned task time to a bus is more complicated as considering the single task time currently to be assigned is not enough to ensure the final bus task time evenly allocated. This is because the allocation of current task may make a bus ending at different shelters causing the remaining task time different. Therefore, the unassigned task time should have two parts, the current task time to be assigned and the subsequent task time to be assigned. To accurately calculate the second part requires considering various possibilities making the problem unnecessarily complicated. To make the algorithm solvable in a polynomial time, in this work, the second part is estimated using the bus capacity shortage of the shelter, *i.e.* $\mathcal{T}_l = t_l + \overline{T}_{s_l}$ is calculated as the unassigned task time to apply Theorem 2. This is because if a bus is allocated with task $\mathcal{T}_l$, it will end at shelter s_l. In the worst case, the bus has to complete the total bus capacity shortage. Therefore, $\overline{T}_{s_l}$ is an upper bound of subsequent task time ending at shelter s_l.

As allocating a task of shelter i to bus $b \in B_i$ does not have any extra travel time, this kind assignment should be prioritized. However, sometimes those buses at shelter i may not have enough $C_b, \forall b \in B_i$ to contain the task time t_l or there is no bus ending at shelter i at all. To ensure the evacuation time do not exceed T_r rapidly, task $\mathcal{T}_l$ will be assigned to a bus $b \in B_j, j \in S$. This means the task $\mathcal{T}_l = \{i, p_l, s_l, t_l\}$ has to change to $\mathcal{T}_l' = \{j, p_l, s_l, t_l'\}$. Since the pickup point and ending shelter are unchanged, this kind of changes to tasks ensure no evacuee omitted and no shelter capacity exceeded. And the new task travel time is determined as, $t_l' = t_l + t_b^l$, where t_b^l indicates the additional travel time needed for this change. Then the changed task $\mathcal{T}_l'$ will be assigned to bus $b = \arg\max_{b \in B} \{C_b\}$ based on Theorem 2.

The more details of the algorithm of task assignment are described in Algorithm 4.

Algorithm 4 Task assignment

```
Input: The list of all the tasks, $\mathcal{T}$;
        $\alpha$ times of the lower bound of evacuation time, $T_r = \max\left\{\alpha\frac{T_t^*}{n_b}, \max_{1\le i\le Q}\mathcal{T}.t_i\right\}$;
        An case of BEP with the locations of buses, $\mathcal{L}_b$;
Output: The rescue plan of buses, $\mathcal{P}$
 1: Assign the tasks starting from the depots to each bus's plan , $\mathcal{P}_b = \{\mathcal{T}_1^b | \mathcal{T}_1^b.d_1^b = \mathcal{L}_b\}, \forall b \in B$
 2: For each bus $b \in B$, $C_b \leftarrow T_r - t_{b_1}$
 3: Update $\mathcal{L}_b = s_1^b$ and Remove assigned tasks from $\mathcal{T}$
 4: while $\mathcal{T} \neq \emptyset$ do
 5:     For each shelter $j \in S$, $\overline{T}_j \leftarrow \sum_{d_l=j} t_l - \sum_{b\in B_j} C_b$
 6:     $l \leftarrow \arg\max_l\{t_l + T_j | d_l = j\}$
 7:     $b \leftarrow \arg\max_b\{C_b | C_b >= t_l \& \mathcal{L}_b = j\}$
 8:     if $b = \emptyset$ then
 9:         $t_b^l \leftarrow$ the additional time if the $d_l$ is changed to $\mathcal{L}_b, \forall b \in B$
10:         $b \leftarrow \arg\max_{b\in B}\{C_b - t_b^l\}$
11:         Change $\mathcal{T}_l.d_l$ to $\mathcal{L}_b$
12:     end if
13:     $\mathcal{P}_b \leftarrow \{\mathcal{P}_b, \mathcal{T}_l\}$
14:     $\mathcal{L}_b \leftarrow s_l^b$
15:     Remove task $l$ from $\mathcal{T}$
16:     $T_r \leftarrow \max(T_r, \{\sum_{1\le k\le m} t_k^b | \forall b \in B\})$
17:     $C_b \leftarrow T_r - \sum_{1\le k\le m} t_k^b, \forall b \in B$
18: end while
19: $\mathcal{P} \leftarrow \{\mathcal{P}_b | \forall b \in B\}$
20: return $\mathcal{P}$
```

IV. Case Studies

The algorithms presented above are coded in MATLAB R2018a and run on a desktop with CPU Intel Core i5-7500 3.20 GHz and 20 GB RAM. The network flow model is solved by CPLEX. In this section, an illustrative case is used to illustrate the BEP and the process of NFP. A number of random cases are used to study the influence of the choice of the parameter α in the post-processing algorithm. To verify the effectiveness of the proposed model and solution, another group of random cases and a real-world disaster, Xingguo flood disaster, occurred in China in 2019 are also studied and illustrated.

A. Illustrative case

A simple case illustrated in Figure 2 is used to explain the problem and the process of the two-step polynomial-time approach. There are two depots, three pickup points and two shelters given. The pickup points have demands of $Q_i = (2, 1, 3)$, while the shelters have capacities of $U_j = (4, 3)$. There are two buses parked in depot D_1, while one bus is located in depot D_2 at the beginning of evacuation. Without loss of generality, the travel time of two directions of one path is different. The red numbers above the lines are the travel time from pickup points to shelters while the green numbers under the lines are the travel time from shelters to pickup points, and the black numbers on the lines are the travel time starting from depots to pickup points.

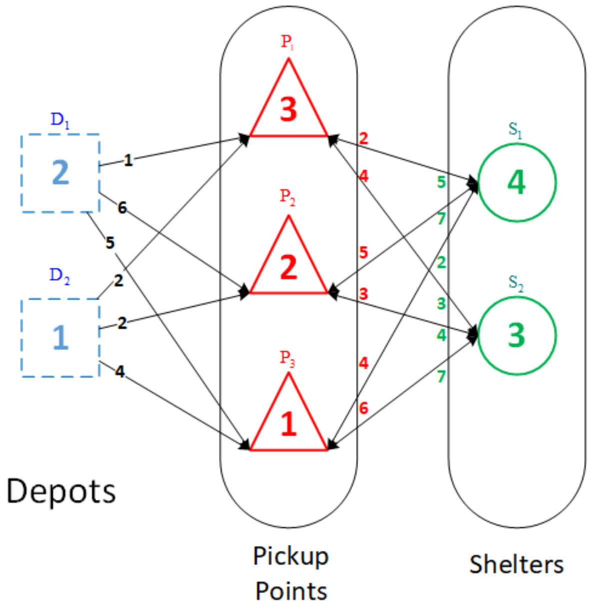


Fig. 2. An illustrative case of BEP

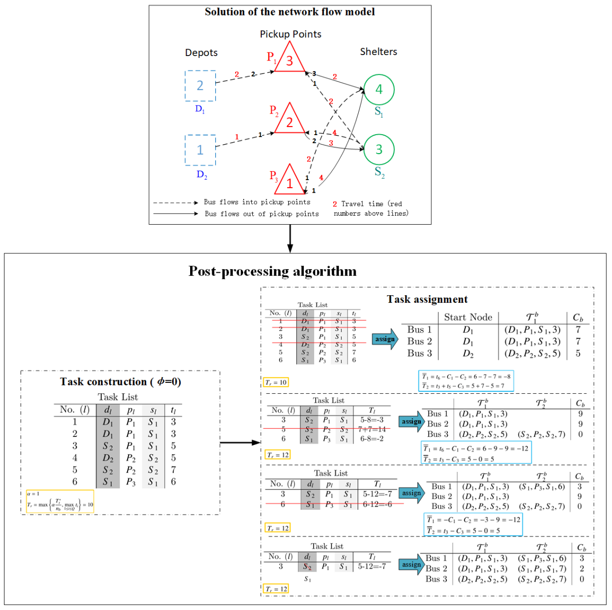
Table I represents an optimal solution of the presented case. Bus 1 starts from depot D_1, and then rescues evacuees from pickup point P_1 to shelter S_1 twice. Its total travel time is thus given by $\delta_{D_1P_1} + \delta_{P_1S_1} + \delta_{S_1P_1} + \delta_{P_1S_1} = 1+2+5+2 = 10$. For Bus 2, the travel time is 9, and for Bus 3, it is 12, resulting in the minimum evacuation time $T_e^* = 12$.

TABLE I
OPTIMAL SOLUTION OF THE ILLUSTRATIVE CASE

	Start Node	$\mathcal{T}_1^b$	$\mathcal{T}_2^b$
Bus 1	D_1	$(D_1, P_1, S_1, 3)$	$(S_1, P_1, S_1, 7)$
Bus 2	D_1	$(D_1, P_1, S_1, 3)$	$(S_1, P_3, S_1, 6)$
Bus 3	D_2	$(D_2, P_2, S_2, 5)$	$(S_2, P_2, S_2, 7)$

Using the NFP proposed, a network flow model is formulated and solved firstly. Figure 3 shows the solution of the network flow model and the evolution of the post-processing algorithm. The minimum total travel time obtained from the network flow model is $T_t^* = 29$. A lower bound of the minimum evacuation time is $\underline{T_e^*} = T_t^*/n_b = \lceil 29/3 \rceil = 10$, which is smaller than the actual minimum evacuation time $T_e^* = 12 \geq \underline{T_e^*}$.

To construct tasks for P_1, 3 inflows, 2 from D_1 and one from S_2, are ranked as D_1, D_1 and S_2 for travel time of 1, 1, and 3. 3 outflows all are to S_1 with travel time of 2. Therefore, 3 constructed tasks are (D_1, P_1, S_1), (D_1, P_1, S_1) and (S_2, P_1, S_1) with travel time of 3, 3, and 5, respectively. For P_2 there are 2 inflows from D_2 and S_2 with travel time of 2 and 4 respectively, and 2 outflows to S_2 with travel time of 3. Hence, two paired tasks are (D_2, P_2, S_2) and (S_2, P_2, S_2) with travel time of 5 and 7, respectively. For P_3, the single inflow from S_1 and single outflow to S_1 are paired as a task (S_1, P_3, S_1) with travel time of 6. These 6 constructed tasks are shown in Figure 3.


Solution of the network flow model
Pickup Points
Depots
Shelters
P_3 3
P_2 2
P_1 1
2 D_1
1 D_2
4 S_1
3 S_2
Bus flows into pickup points
Bus flows out of pickup points
2 Travel time (red numbers above lines)
Post-processing algorithm
Task construction (ϕ=0)
Task assignment
assign

Task List

No. (l)	d_l	p_l	s_l	t_l
1	D_1	P_1	S_1	3
2	D_1	P_1	S_1	3
3	S_2	P_1	S_1	5
4	D_2	P_2	S_2	5
5	S_2	P_2	S_2	7
6	S_1	P_3	S_1	6

$\alpha = 1$
$T_r = \max\left\{\alpha \frac{T_t^*}{n_b}, \max_{1\leq i\leq Q} t_i\right\} = 10$

Task List

No. (l)	d_l	p_l	s_l	t_l
~~1~~	~~D_1~~	~~P_1~~	~~S_1~~	~~3~~
~~2~~	~~D_1~~	~~P_1~~	~~S_1~~	~~3~~
3	S_2	P_1	S_1	5
~~4~~	~~D_2~~	~~P_2~~	~~S_2~~	~~5~~
5	S_2	P_2	S_2	7
6	S_1	P_3	S_1	6

	Start Node	$\mathcal{T}_1^b$	C_b
Bus 1	D_1	$(D_1, P_1, S_1, 3)$	7
Bus 2	D_1	$(D_1, P_1, S_1, 3)$	7
Bus 3	D_2	$(D_2, P_2, S_2, 5)$	5

$T_r = 10$
$\overline{T}_1 = t_6 - C_1 - C_2 = 6 - 7 - 7 = -8$
$\overline{T}_2 = t_3 + t_5 - C_3 = 5 + 7 - 5 = 7$

Task List

No. (l)	d_l	p_l	s_l	T_l
3	S_2	P_1	S_1	5-8=-3
~~5~~	~~S_2~~	~~P_2~~	~~S_2~~	~~7+7=14~~
6	S_1	P_3	S_1	6-8=-2

	$\mathcal{T}_1^b$	$\mathcal{T}_2^b$	C_b
Bus 1	$(D_1, P_1, S_1, 3)$		9
Bus 2	$(D_1, P_1, S_1, 3)$		9
Bus 3	$(D_2, P_2, S_2, 5)$	$(S_2, P_2, S_2, 7)$	0

$T_r = 12$
$\overline{T}_1 = t_6 - C_1 - C_2 = 6 - 9 - 9 = -12$
$\overline{T}_2 = t_3 - C_3 = 5 - 0 = 5$

Task List

No. (l)	d_l	p_l	s_l	T_l
3	S_2	P_1	S_1	5-12=-7
~~6~~	~~S_1~~	~~P_3~~	~~S_1~~	~~6-12=-6~~

	$\mathcal{T}_1^b$	$\mathcal{T}_2^b$	C_b
Bus 1	$(D_1, P_1, S_1, 3)$	$(S_1, P_3, S_1, 6)$	3
Bus 2	$(D_1, P_1, S_1, 3)$		9
Bus 3	$(D_2, P_2, S_2, 5)$	$(S_2, P_2, S_2, 7)$	0

$T_r = 12$
$\overline{T}_1 = -C_1 - C_2 = -3 - 9 = -12$
$\overline{T}_2 = t_3 - C_3 = 5 - 0 = 5$

Task List

No. (l)	d_l	p_l	s_l	T_l
3	$\cancel{S_2}$ S_1	P_1	S_1	5-12=-7

	$\mathcal{T}_1^b$	$\mathcal{T}_2^b$	C_b
Bus 1	$(D_1, P_1, S_1, 3)$	$(S_1, P_3, S_1, 6)$	3
Bus 2	$(D_1, P_1, S_1, 3)$	$(S_1, P_1, S_1, 7)$	2
Bus 3	$(D_2, P_2, S_2, 5)$	$(S_2, P_2, S_2, 7)$	0

$T_r = 12$

Fig. 3. The solution of the network flow model and the evolution of the post-processing algorithm for the illustrative case

Due to the starting nodes of the buses, Task 1,2,4 are assigned to the three buses respectively and C_b of the buses are 7,7,5, for $T_r = \max\left\{\alpha \frac{T_t^*}{n_b}, \max_{1\leq i\leq Q} t_i\right\} = 10$, *i.e.* $\alpha = 1$. After these assignments, Bus 1 and 2 are ending at S_1, while Bus 3 is ending at S_2. But Task 3 and 5 start from S_2 and only Task 6 starts from S_1. Therefore, $\overline{T}_1 = 6-(7+7) = -8$, $\overline{T}_2 = 5+7-5 = 7$ and the predicted travel time of Task 3, 5, 6 are respectively -3, 14, -2. This leads to Task 5 to be assigned next. Since only Bus 3 is at S_2, $C_3 < T_5$, Bus 3 is the bus that causes the minimum increase of T_r after accepting Task 5. After Task 5 is assigned, T_r increases to 12, C_b of buses are updated to 9,9,0 and $\overline{T}_1 = -12, \overline{T}_2 = 5$. Then Task 6 is assigned to Bus 1 within its remaining capacity. Finally, Task 3 starts from S_2, and only Bus 3 is there. Since Bus 3 has no capacity left, the task is re-routed from S_1 as (S_1, P_1, S_1) resulting in a travel time of 7. Since the new travel time is less than $C_2 = 9$, the new Task 3 is assigned to Bus 2. This completes the task assignment and the final evacuation time is 12, which is also optimal.

B. *Parameter tuning of α in post-processing algorithm*

To some extent, the performance of post-processing algorithm depends upon the suitable selection of the value of α in task assignment. In task assignment, the reference and guidance of the evacuation time that can be achieved is associated with α times of the lower bound of the minimum evacuation, *i.e.*, $T_r = \max\left\{\alpha \frac{T_t^*}{n_b}, \max_{1\leq i\leq Q} t_i\right\}$. If α is more close to 1, the bus remaining capacity C_b will run out more quickly so that task adjustments have to be carried out earlier and more frequently resulting in a higher evacuation time. On the other side, if α is larger, C_b is larger, hence the final evacuation time will also be larger.

To fine the most suitable value of α, 200 random cases are generated with the number of pickup points and shelters ranging of $[2, 41]$. Values of δ_{ij}, U_j are drawn randomly in the range of $[1, 10]$ and Q_i is in the range of $[1, 5]$. The number of available buses is drawn randomly in the range of $[1, Q]$, which means all the buses are expected to accept at least one rescue task. To tune α, every one of 200 random cases is calculated with 50 values of α ranged from 1 to 1.98.

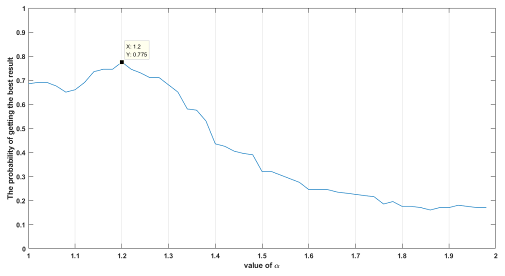


Fig. 4. The probability of getting the best result with different α

To evaluate the performance of different values of α, for each case, the values of α that make the evacuation time equal the minimum evacuation time of all the 50 results are recorded. And shown in Figure 4, about 78% cases achieve the best result among 50 results of that case when $\alpha = 1.2$. Therefore, in the following studies, α is set up as 1.2, namely, $T_r = \max\left\{1.2\underline{T_e^*}, \max_{1\leq i\leq Q} t_i\right\}$.

C. *Comparing polynomial-time approach with other algorithms*

To illustrate the result of NFP, 9 sets of 10, total 90 random cases with varying size are generated and studied. Following the set up designed by Goerigk et al. [21], values δ_{ij}, U_j are drawn randomly in the range of $[1, 10]$ and Q_i in the range of $[1, 5]$. Table II gives an overview of the respective values for the number of demand point n_p, the number of shelters n_s, the number of depot n_d and the number of buses n_b.

TABLE II
RANDOMIZED CASE SIZES

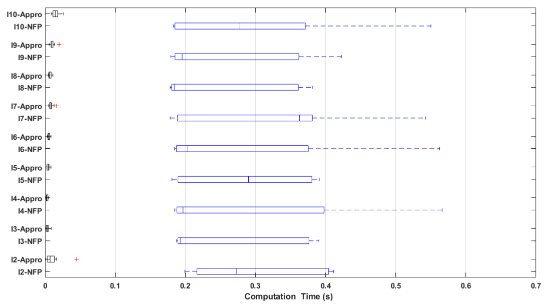
Set Name	n_d	n_s	n_p	n_b
I_2	1	2	2	2
I_3	1	3	3	2
I_4	1	4	4	3
I_5	1	5	5	3
I_6	1	6	6	4
I_7	1	7	7	4
I_8	1	8	8	5
I_9	1	9	9	5
I_{10}	1	10	10	6

To test the performance of the proposed two-step polynomial-time approach, random cases are solved by NFP and the computation time is recorded. Then the results are compared with the performance of CPLEX, genetic algorithm(GA) and approximation algorithm proposed in Pedrosa and Schouery [20]. To compare the efficiency of different algorithms, the computation time of NFP is also the time limitation of CPLEX or GA. When CPLEX or GA did not find the optimal solutions on time, the best solutions found are given. And population size, crossover probability, mutation probability and maximum number of generation of GA are chosen at 60, 90%, 2% and 200, respectively. The computation times of NFP and approximation algorithm are shown in Figure 5(a). It is noticed that though the computation time of NFP is a little more than that of approximation algorithm, it is still less than 0.6 seconds for most cases and the thin margins are negligible in any real-world cases.

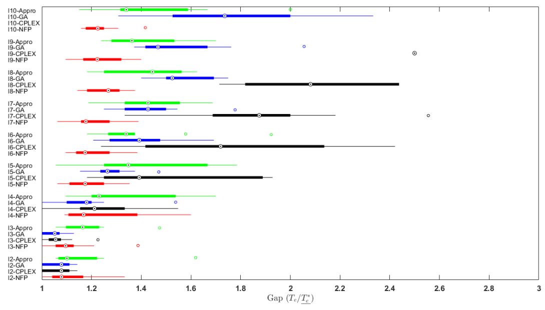
In addition, results of all the algorithms are shown in Figures 5(b). Within that short computation time, the results of NFP are mostly under 1.4 times of $\underline{T_e^*} = T_t^*/n_b$, while the gaps of CPLEX, GA and approximation algorithm surge considerably with the size of case growth. Moreover, on Set I_9 and I_{10} CPLEX has difficulties in producing results, consuming lots of memory resources and using the whole time in finding a solution in that short time. In general, for small size cases, such as I_2, $I_3$and I_4, the GA and CPLEX approaches are slightly better than the proposed NFP. However, as the size of cases increases, the superiority of NFP becomes clearer. Therefore, the proposed NFP approach stands out of other algorithms to closely approximate the minimum evacuation time for large-scale disasters.

D. Xingguo flood disaster

In order to examine the performance of the proposed algorithm in a real situation, the flood disaster in Xingguo, China in 2019 is studied. The disaster affected approximately 412,600 people, from which around 66,000 were taken to shelters. Figure 6 shows the proposed scenario. In order to solve the real scenario using the proposed BEP model, the following assumptions are considered:



(a) Computation time comparison of the approximation algorithm and NFP



(b) Boxplot of scores per case set for NFP, CPLEX, GA and approximation algorithm

Fig. 5. Comparison of NFP, CPLEX, GA and approximation algorithm

- The shelters and their capacities were estimated according to local government reports. Shown in Table IV, 25 shelters with a total capacity of 408,000 were considered.
- The number of evacuees in each town was estimated considering the disaster situation and population density of the area. And the total number of evacuees in Xingguo is 66,000 distributed among 22 villages. Table III shows the population of various villages that need to be evacuated.
- For the collection points, the bus stops nearest to the center of each risky town are used to ensure that the buses were able to reach the evacuees. Due to the administrative division of the zone, 22 collection points were considered.
- A depot is set at the largest and highest-level shelter near the endangered area.
- Due to the large number of evacuees in each collection point, the fleets of public transport have been considered, which are 10 buses with a total capacity for 500 passengers in a fleet. Taking the buses in the same fleet as a whole, the number of fleets considered goes from 15 to 140 fleets. All the buses start from the depot.
- The travel time between nodes was calculated by Bioinformatics Toolbox of MATLAB assuming the speed of buses is from 20 km/h to 120 km/h which depends on the damage and level of the road. The travel time of the road network is asymmetrical.

TABLE III
INFORMATION OF EVAUEES FOR XINGGUO FLOOD DISASTER

Village no.	Name	Evacuees	Q_i
1	Shefu	1437	3
2	Longkou	2685	6
3	Jiecun	813	2
4	Yongfeng	491	1
5	Butou	6462	13
6	Gulonggang	2949	6
7	Jiangbei	3921	8
8	Zhangmu	654	2
9	Longping	492	1
10	Dongcun	4093	9
11	Lianjiang	6756	14
12	Juncun	493	1
13	Denglong	8160	17
14	Xinglian	1582	4
15	Chayuan	1392	3
16	Meijiao	13394	27
17	Fangtai	231	1
18	Gaoxing	5255	11
19	Chengang	785	2
20	Xingjiang	1766	4
21	Liangcun	2110	5
22	Chongxian	756	2

TABLE IV
CAPACITY OF SHELTERS FOR XINGGUO FLOOD DISASTER

Shelter No.	Capacity	U_j
1	6500	13
2	2500	5
3	2000	4
4	10760	21
5	4494	8
6	7000	14
7	4500	9
8	7700	15
9	5000	10
10	46200	92
11	4000	8
12	124666	249
13	50000	100
14	3000	6
15	5100	10
16	11350	22
17	1500	3
18	9800	19
19	23345	46
20	35684	71
21	4000	8
22	7500	15
23	26000	52
24	5000	10
25	500	1

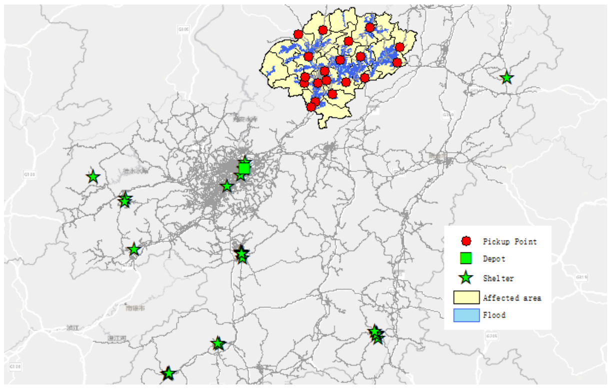
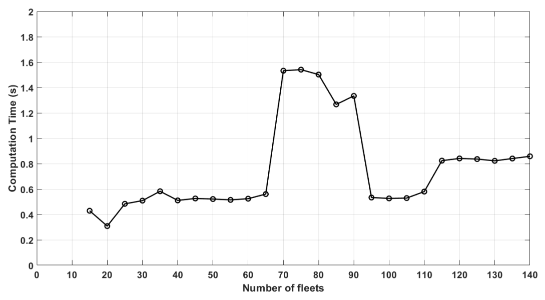
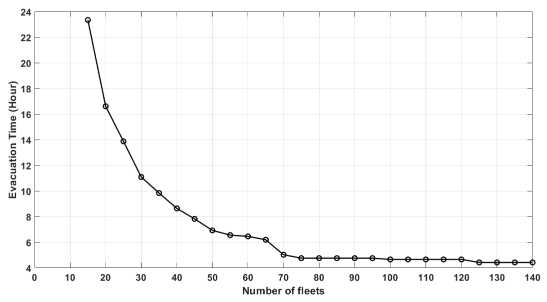


Fig. 6. Evacuation scenario for the great flood disaster of Xingguo in 2019

The influence of the number of used fleets is evaluated in this study. Figures 7(a) and 7(b) show the computation time and the evacuation time obtained for each case. The computation time to obtain the evacuation plan is less than 1.5 seconds on the Xingguo flood disaster, which is highly acceptable for finding a reasonable evacuation plan in reality. Results show that the evacuation time decreases with the amount of fleets from nearly one day to 4 hours, and the shape is a convex curve. Like established in Bish [29], the evacuation time improves slightly when the number of fleets is more than 75.



(a) Computation time of Xingguo flood disaster with varied numbers of fleets



(b) Evacuation time for different numbers of fleets

Fig. 7. Results of Xingguo flood disaster

V. CONCLUSION

In this paper, a rigorous formulation of the Bus Evacuation Problem, inspired by that given in Goerigk et al. [21], is presented. As the time consumed by the whole evacuation is a priority considered in disaster management, the objective of BEP is to minimize the evacuation time. However, this problem is NP-hard. And for large-scale disasters, the 0-1 integer linear programming model is hard to be solved in a short time and limited memory resources. Therefore, in this paper, a two-step polynomial-time approach called network flow planning (NFP) approach is proposed for large-scale disasters. The variables of 0-1 integer linear programming model are aggregated over all the buses and all trips resulting in a network flow model with the objective of minimizing the total travel time of all the buses. The network flow model is proven solvable with a linear programming algorithm in polynomial time. To transform the solution of network flow model into a reasonable evacuation plan and approximate the minimum evacuation time, a post-processing algorithm for the network flow model is designed. In the post-processing algorithm, the solution of the network flow model is transferred into a task list by pairing the in- and out- flows of each pickup point first. Then all the tasks are assigned to available buses as even as possible. A lower bound of the minimum evacuation time is obtained from dividing the minimum total travel time by the number of available buses, and a reference related to the lower bound is considered during the post-processing algorithm.

To test NFP, two sets of cases are proposed: the first data set is built with random values with the same sizes of that proposed by Goerigk et al. [21], and the second one is based on a real-world scenario of a recent flood disaster occurring in Xingguo, China. Compared with GA and the commercial general integer programming solver, CPLEX, on the same computation time, the implemented approach has an outstanding performance for large size cases. The computation time of the whole NFP approach is a slightly larger than that of approximation algorithm proposed in Pedrosa and Schouery [20], but it is still less than 0.6 seconds in most cases and its performance is much better and more stable than that of approximation algorithm in large-scale cases. In addition, for the Xingguo flood disaster problem, which has 22 pickup points and 25 shelters, the proposed NFP is able to find reasonable evacuation plans within 2 seconds of computation time. And the results show that with the number of rescue bus fleets increasing, the evacuation time needed dwindles firstly and stabilizes at 4.75 hours after the bus fleets is more than 80, which is similar to the actual circumstance.

In practical applications of this two-step polynomial-time approach, it is essential to collect the accurate relevant data in advance, including evacuation demand, available buses, travel time of each path, etc. Hence, effective information acquisition and uncertainties of data deserve further studies. Particularly, in future work, the uncertainties in disaster information will be considered and the evacuation plans will be updated in time.

Acknowledgment

This work was supported by National Key Research and Development Project of China (No.2017YFC1502902)

References

[1] X. Yang, X. Ban, and J. Mitchell, "Modeling multimodal transportation network emergency evacuation considering evacuees' cooperative behavior," *Transportation Research Part a-Policy and Practice*, vol. 114, pp. 380–397, 2018.

[2] E. Pourrahmani, M. R. Delavar, P. Pahlavani, and M. A. Mostafavi, "Dynamic evacuation routing plan after an earthquake," *Natural Hazards Review*, vol. 16, no. 4, 2015.

[3] W. Yi and A. Kumar, "Ant colony optimization for disaster relief operations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 660–672, 2007.

[4] V. Krasko and S. Rebennack, "Two-stage stochastic mixed-integer nonlinear programming model for post-wildfire debris flow hazard management: Mitigation and emergency evacuation," *European Journal of Operational Research*, vol. 263, no. 1, pp. 265–282, 2017.

[5] V. C. Pereira and D. R. Bish, "Scheduling and routing for a bus-based evacuation with a constant evacuee arrival rate," *Transportation Science*, vol. 49, no. 4, pp. 853–867, 2015.

[6] B. H. Oh, K. Kim, H.-L. Choi, and I. Hwang, "Cooperative multiple agent-based algorithm for evacuation planning for victims with different urgencies," *Journal of Aerospace Information Systems*, vol. 15, no. 6, pp. 382–395, 2018.

[7] M. Yusoff, J. Ariffin, and A. Mohamed, "Dpso based on a min-max approach and clamping strategy for the evacuation vehicle assignment problem," *Neurocomputing*, vol. 148, pp. 30–38, 2015.

[8] S. Shahparvari, B. Abbasi, P. Chhetri, and A. Abareshi, "Fleet routing and scheduling in bushfire emergency evacuation: A regional case study of the black saturday bushfires in australia," *Transportation Research Part D: Transport and Environment*, 2017.

[9] S. Shahparvari, B. Abbasi, and P. Chhetri, "Possibilistic scheduling routing for short-notice bushfire emergency evacuation under uncertainties: An australian case study," *Omega-International Journal of Management Science*, vol. 72, pp. 96–117, 2017.

[10] G. Ozbaygin, O. Karasan, and H. Yaman, "New exact solution approaches for the split delivery vehicle routing problem," *EURO Journal on Computational Optimization*, vol. 6, no. 1, pp. 85–115, 2017.

[11] M. Heydar, J. Yu, Y. Liu, and M. E. H. Petering, "Strategic evacuation planning with pedestrian guidance and bus routing: a mixed integer programming model and heuristic solution," *Journal of Advanced Transportation*, vol. 50, no. 7, pp. 1314–1335, 2016.

[12] F. Sabouhi, A. Bozorgi-Amiri, M. Moshref-Javadi, and M. Heydari, "An integrated routing and scheduling model for evacuation and commodity distribution in large-scale disaster relief operations: a case study," *Annals of Operations Research*, 2018.

[13] H. Zheng, "Optimization of bus routing strategies for evacuation," *Journal of Advanced Transportation*, vol. 48, no. 7, pp. 734–749, 2014.

[14] L. Özdamar and O. Demir, "A hierarchical clustering and routing procedure for large scale disaster relief logistics planning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 591–602, 2012.

[15] J.-B. Sheu and C. Pan, "A method for designing centralized emergency supply to respond to large-scale natural disasters," *Transportation Research Part B-Methodological*, vol. 67, pp. 284–305, 2014.

[16] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans Cybern*, vol. 46, no. 3, pp. 582–94, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/25794408

[17] Lakshay and N. B. Bolia, "Operating strategies of buses for mass evacuation," *Safety Science*, vol. 111, pp. 167–178, 2019.

[18] Z. M. Huang, W. N. Chen, Q. Li, X. N. Luo, H. Q. Yuan, and J. Zhang, "Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation," *IEEE Trans Cybern*, vol. PP, 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/32915756

[19] M. Goerigk and B. Grün, "A robust bus evacuation model with delayed scenario information," *OR Spectrum*, vol. 36, no. 4, pp. 923–948, 2014.

[20] L. L. C. Pedrosa and R. C. S. Schouery, "Approximation algorithms for the bus evacuation problem," *Journal of Combinatorial Optimization*, vol. 36, no. 1, pp. 131–141, 2018.

[21] M. Goerigk, B. Grün, and P. Hebler, "Branch and bound algorithms for the bus evacuation problem," *Computers and Operations Research*, vol. 40, no. 12, pp. 3010–3020, 2013.

[22] M. Goerigk, B. Grün, and P. Heßler, "Combining bus evacuation with location decisions: A branch-and-price approach," *Transportation Research Procedia*, vol. 2, pp. 783–791, 2014.

[23] A. Kulshrestha, Y. Lou, and Y. Yin, "Pick-up locations and bus allocation for transit-based evacuation planning with demand uncertainty," *Journal of Advanced Transportation*, vol. 48, no. 7, pp. 721–733, 2014.

[24] M. Niendorf and A. R. Girard, "Exact and approximate stability of solutions to traveling salesman problems," *IEEE Trans Cybern*, vol. 48, no. 2, pp. 583–595, 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/28103566

[25] M. Goerigk, K. Deghdak, and V. T'Kindt, "A two-stage robustness approach to evacuation planning with buses," *Transportation Research Part B: Methodological*, vol. 78, pp. 66–82, 2015.

[26] M. Goerigk, K. Deghdak, and P. Heßler, "A comprehensive evacuation planning model and genetic solution algorithm," *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 82–97, 2014.

[27] S. Shahparvari and B. Abbasi, "Robust stochastic vehicle routing and scheduling for bushfire emergency evacuation: An australian case study," *Transportation Research Part A: Policy and Practice*, vol. 104, pp. 32–49, 2017.

[28] G. Dikas and I. Minis, "Solving the bus evacuation problem and its variants," *Computers and Operations Research*, vol. 70, pp. 75–86, 2016.

[29] D. R. Bish, "Planning for a bus-based evacuation," *OR Spectrum*, vol. 33, no. 3, pp. 629–654, 2011.

[30] A. Ghouila-Houri, "Caractérisation des matrices totalement unimodulaires," *C. R. Acad. Sci. Paris*, vol. 254, pp. 1192–1194, 1962.

[31] A. Hoffman and J. Kruskal, *Integral Boundary Points of Convex Polyhedra*, 11 2010, vol. 38, pp. 49–76.

[32] S. Chubanov, "A strongly polynomial algorithm for linear systems having a binary solution," *Mathematical Programming*, vol. 134, no. 2, pp. 533–570, 2012. [Online]. Available: https://doi.org/10.1007/s10107-011-0445-3