

Research Article

Network Reconfiguration with Orientation-Dependent Transit Times

Hari Nandan Nath,¹ Urmila Pyakurel ,² and Tanka Nath Dhamala ²

¹*Bhaktapur Multiple Campus, Department of Mathematics, Tribhuvan University, Bhaktapur, Nepal*

²*Central Department of Mathematics, Tribhuvan University, P. O. Box 13143, Kathmandu, Nepal*

Correspondence should be addressed to Urmila Pyakurel; urmilapyakurel@gmail.com

Received 5 December 2020; Revised 6 February 2021; Accepted 23 February 2021; Published 11 March 2021

Academic Editor: Fernando Bobillo

Copyright © 2021 Hari Nandan Nath et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Motivated by applications in evacuation planning, we consider a problem of optimizing flow with arc reversals in which the transit time depends on the orientation of the arc. In the considered problems, the transit time on an arc may change when it is reversed, contrary to the problems considered in the existing literature. Extending the existing idea of auxiliary network construction to allow asymmetric transit time on arcs, we present strongly polynomial time algorithms for solving single-source-single-sink maximum dynamic contraflow problem and quickest contraflow problem. The results are substantiated by a computational experiment in a Kathmandu road network. An algorithm to solve the corresponding earliest arrival contraflow problem with a pseudo-polynomial-time complexity is also presented. The partial contraflow approach for the corresponding problems has also been discussed.

1. Introduction

The transportation network is represented as a dynamic network in which road segments represent the arcs and intersection between them are the nodes. The unsafe locations of network are the sources and the locations at safe regions are the sinks. Each node and arc of the network are bounded by finite capacities. Each arc has a transit time or a cost function that determines the amount of time or cost needed to travel it. The network may contain arcs in both directions with different capacities and asymmetric transit times or costs. The computational complexities for the transportation planning are heavily dependent upon the number of sources, sinks, parameters on the arcs, and nodes, like constant, time-dependent, or flow-dependent capacities or transit times as well as additional constraints. The time we consider mostly is discrete, which approximates the computationally heavy continuous models at the cost of solution approximations. Also, the constant time is probably approximated by free flow speeds or certain queuing rules and constant capacity settings that reduce the problem to be linear, at least more tractable, in contrast to

the more general and realistic flow-dependent traffic flow scenarios. For extensive explanation on diversified theories and applications, we recommend Dhamala et al. [1] and Kotsireas et al. [2] and the citations therein.

The transportation network, during or after disastrous situations, becomes more congested due to large number of people and vehicles towards the safer areas on the streets. Moreover, the movement towards risk areas from safer places is discouraged because of which the corresponding road lanes are almost empty. The empty lanes management plays vital role to reduce the traffic congestion. The optimal lane reversal strategy makes the traffic systematic and smooth by removing the traffic jams caused in different large-scale natural and man-made disasters, busy office hours, special events, and street demonstrations. The contraflow reconfiguration, by means of various operations research models, heuristics, optimization, and simulation techniques, reverses the usual direction of empty lanes towards the sinks satisfying the given constraints that increase the flow value and decrease the average evacuation time [3, 4].

Although the capacity of lanes is assumed to be constant, the transit times may vary. If we consider the inflow-dependent or load-dependent transit times, the empty lanes may have likely zero time contrary to the congested segments where the transit times increase with the flow density. In reality, the transit time of incoming lanes towards the sources may not be equal to the outgoing lanes. It may be the cases depending on the network topology as well. In such cases, the free flow transit time has been considered for the reversed lanes [4]. Recently, the authors in [5] have considered the asymmetric transit times on the lanes but those lanes are individual and reversal decisions are made with their own transit times, that is, symmetric transit times such as in [3, 6, 7].

In this paper, we consider the asymmetric transit times of reversed lanes in general form and show their impact on optimal lane reversals; that is, our transit time is dependent on the orientation of the lanes which applies asymmetric times. The capacities of both reversals are the same by adding both capacities.

Consider a network in Figure 1(a). The arc labels represent the capacity of the arc and travel time on the direction of the arc. For example, the capacity of (s, a) is 4 and its travel time is 1. That means a flow of 4 units per unit time can be sent from s to a and takes 1 unit of time to reach from s to a . For a time horizon of 6 units, a maximum of 8 units of flow (1 along $s-a-d$ twice, 1 along $s-b-d$ thrice, and 1 along $s-a-b-d$ thrice) can be sent from s to d without allowing arc reversals.

If (b, a) is reversed and the transit time is kept intact (Figure 1(b)), the maximum of 10 units of flow (1 along $s-a-d$ twice, 1 along $s-b-d$ thrice, 1 along $s-a-b-d$ (that contains original (a, b)) thrice, and 2 along $s-a-b-d$ (that contains (b, a) reversed) once) can be sent.

If the transit time depends on the orientation, when (b, a) is reversed, its transit time becomes that of original (a, b) (Figure 1(c)). In this case, there is a maximum of 14 units of flow (1 along $s-a-d$ twice, 1 along $s-b-d$ thrice, and 3 along $s-a-b-d$ thrice). In this case, we can add the capacities of (a, b) , (b, a) and replace the two arcs by a single arc (a, b) . If (a, b) is reversed, however, the maximum flow value reduces to 5 (Figure 1(d)).

Kim et al. [8] firstly model the contraflow problem as an integer programming problem, thereby proving its \mathcal{NP} -hardness. As finding exact mathematical solutions for general contraflow techniques is costly, they present two greedy and bottleneck heuristics for possible numerical approximate solutions to the quickest contraflow problem. With computational experiments, it has been shown that at least 40% evacuation time can be reduced by reverting at most 30% arcs. Rebennack et al. [3] solve the two-terminal maximum and quickest contraflow problems optimally in strongly polynomial times. Pyakurel and Dhamala [6] solve the earliest arrival contraflow problem on a two-terminal network in pseudo-polynomial-time. They solve it in strongly polynomial time if the network is series-parallel. The continuous time solution is given in [7] and the solutions with similar objectives are given in [9, 10]. They show that if the minimum cut arcs have symmetric capacities, then the flow is double with the contraflow. Pyakurel et al. [11]

give the first temporally repeated flow algorithm to solve the quickest contraflow problem, within a complexity of solving a min-cost flow problem. The costs for arc reversals as switching costs are studied in [12]. The contraflow models with intermediate storage are introduced in [13].

In this paper, we introduce the maximum dynamic contraflow, earliest arrival, and quickest contraflow problems on asymmetric transit time network and present efficient algorithms to solve these problems in two-terminal networks. Modifying the network transformation suggested by Rebennack et al. [3] in case of symmetric travel time cases, we show that the approach works equally well in the asymmetric travel time settings. The results are extended with partial lane reversals as well. We analyze our solutions in both discrete and continuous time settings. The novelty of this work is to optimize network topology with unequal transit times on reversal arcs to improve congestion by increasing the flow value and decreasing the evacuation time.

We organize the paper as follows. Section 2 presents mathematical formulations and models of the problems allowing asymmetric transit times on arcs. We investigate the maximum dynamic, earliest arrival, and quickest flow problems in Section 3. These results are extended with partial lane reversals in Section 4. In Section 5, some computational results, taking a part of Kathmandu road network as an example network, are presented. The paper is concluded in Section 6.

2. Basic Concepts

A network N is a directed graph consisting of a finite set of nodes V and a finite set of arcs A with $|V| = n$, $|A| = m$. An arc $e \in A$ is associated with a unique pair of nodes i, j : one of them is called the head of e and the other the tail. If e has head i and tail j , then it is called directed from i to j . We consider something known as flow that moves from a set of nodes $S \subset V$, called sources to $D \subset V$ ($S \neq D$). The amount of flow is limited by a capacity function $u: A \rightarrow \mathbb{R}_{\geq 0}$. A travel time $\tau: A \rightarrow \mathbb{R}_{\geq 0}$ is associated with the flow. We denote such a network by $N = (V, A, u, \tau, S, D)$. If $S = \{s\}$ and $D = \{d\}$, we denote the network by $N = (V, A, u, \tau, s, d)$. We define the set of arcs incoming to node i as

$$A_i^- = \{e \in A: i \text{ is the head of } e\}, \quad (1)$$

and the set of arcs going out of it as

$$A_i^+ = \{e \in A: i \text{ is the tail of } e\}. \quad (2)$$

2.1. Static Flow. A static flow is a function $x: A \rightarrow \mathbb{R}_{\geq 0}$ which satisfies the following conditions:

$$\sum_{e \in A_i^+} x(e) - \sum_{e \in A_i^-} x(e) = 0 \quad \forall i \in V \setminus \{S, D\}, \quad (3)$$

$$x(e) \leq u(e) \quad \forall e \in A. \quad (4)$$

The value of the static flow x is

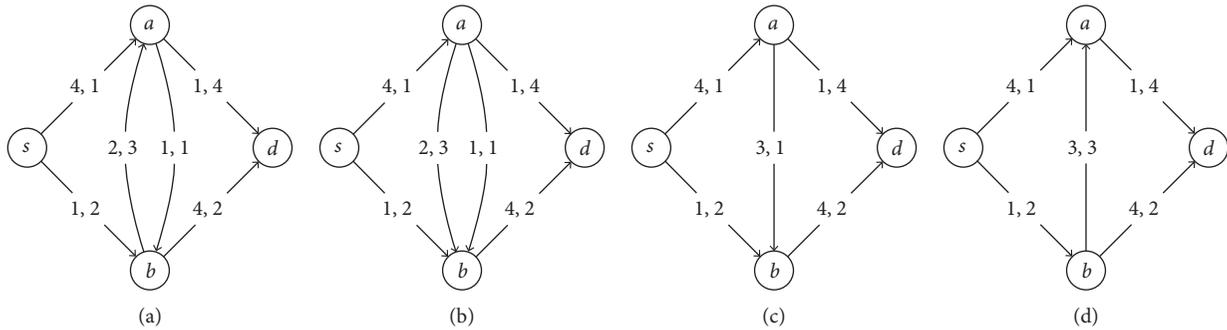


FIGURE 1: (a) Network with arc labels capacity and transit time; (b) (b, a) reversed with the same transit time; (c) (b, a) reversed with orientation-dependent transit time; (d) (a, b) reversed with orientation-dependent transit time.

$$v(x) = \sum_{d \in D} \sum_{e \in A_d^-} x(e). \tag{5}$$

A static flow x is called a circulation if (3) is satisfied by all $i \in V$. A flow x that maximizes $v(x)$ is called a maximum static flow.

2.1.1. Flow Decomposition. Let \mathcal{P} denote the set of all simple paths from S to D , and let \mathcal{C} denote the set of all simple cycles in N . Then every static flow x has a flow decomposition $(x_p)_{p \in \mathcal{P} \cup \mathcal{C}}$ such that $x_p \geq 0 \forall p \in \mathcal{P} \cup \mathcal{C}$ and

$$\sum_{p \in \mathcal{P} \cup \mathcal{C}: x_p: e \in p} x_p = x(e). \tag{6}$$

2.1.2. Minimum Cost Flow. Given two functions $b: V \rightarrow \mathbb{R}$, called supply, and $c: A \rightarrow \mathbb{R}$, called cost with $\sum_{i \in V} b(i) = 0$, a flow x , satisfying (4) and

$$\sum_{e \in A_i^+} x(e) - \sum_{e \in A_i^-} x(e) = b(i) \quad \forall i \in V, \tag{7}$$

is called a minimum cost flow if it minimizes $\sum_{e \in A} c(e)x(e)$. In various applications, the travel time τ is considered as cost.

2.1.3. Residual Network. A very important notion for various network flow calculations is a residual network. Given a static flow x , the residual network N_x has the same vertex set V . The arc set A_x consists of arcs constructed in the following way: For each $e \in A$ directed from i to j , if $x(e) < u(e)$, there is an arc in A_x directed from i to j with residual capacity $u(e) - x(e)$ and cost $c(e)$. If $x(e) > 0$, we have an arc in A_x directed from j to i with residual capacity $x(e)$ and cost $-c(e)$.

For more details, we refer the reader to Ahuja et al. [14].

2.2. Dynamic Flow. A dynamic flow Φ with time horizon T consists of Lebesgue-integrable functions $\Phi_e: [0, T] \rightarrow \mathbb{R}_{\geq 0}$ for each arc $e \in A$ such that $\Phi_e(\theta) = 0$ for $\theta \geq T - \tau(e)$. Here, $\Phi_e(\theta)$ can be realized as the rate of flow entering e at time θ . The flow entering the tail i of the arc $e = (i, j)$ at time θ reaches the

head j of e at time $\theta + \tau_e$. For each $i \in V$, we define the excess of node i induced by Φ at time θ as

$$\text{exc}_\Phi(i, \theta) = \sum_{e \in A_i^-} \int_0^{\theta - \tau(e)} \Phi_e(\sigma) d\sigma - \sum_{e \in A_i^+} \int_0^\theta \Phi_e(\sigma) d\sigma, \tag{8}$$

which is the net amount of flow that is stored at node i up to time θ . In what follows, we assume $S = \{s\}, D = \{d\}$ for simplicity. A feasible dynamic flow Φ satisfies

$$\text{exc}_\Phi(i, \theta) \geq 0 \quad \forall \theta \in [0, T), \forall i \in V \setminus \{s\}, \tag{9}$$

$$\text{exc}_\Phi(i, T) = 0, \quad \forall i \in V \setminus \{s, d\}, \tag{10}$$

$$0 \leq \Phi_e(\theta) \leq u(e), \quad \forall e \in A, \theta \in [0, T). \tag{11}$$

The value of the dynamic flow Φ at time θ is

$$v_\theta(\Phi) = \text{exc}_\Phi(d, \theta), \tag{12}$$

and the total value of the dynamic flow Φ is

$$v(\Phi) = v_T(\Phi) = \text{exc}_\Phi(d, T). \tag{13}$$

For more details, refer to Skutella [15].

Given a time horizon T , the dynamic flow Φ that maximizes $v_T(\Phi)$ is called the *maximum dynamic flow*. Given a flow value Q , the dynamic flow with minimum time horizon T^* such that $v_{T^*}(\Phi) = Q$ is called the *quickest flow*, and the dynamic flow Φ which maximizes $v_\theta(\Phi)$ for all $\theta \in [0, T]$ is called the *earliest arrival flow*.

2.2.1. Temporally Repeated Flow. Given a static flow x and a time horizon T , a flow decomposition on x gives a set of paths \mathcal{P} with flow x_p for each $P \in \mathcal{P}$. Flow is sent along P at a constant rate x_p from time 0 to $\max\{T - \tau(P), 0\}$, where $\tau(P) = \sum_{e \in P} \tau(e)$ is the travel time on path P , to define a dynamic flow known as the temporally repeated flow. To give an explicit expression for the dynamic flow, we define the following for e in a path P directed from i to j :

P_{si} = the portion of the path P from s to i ,

P_{jd} = the portion of the path P from j to d ,

$$\mathcal{P}_e(\theta) = \{P \in \mathcal{P}: e \in P, \tau(P_{si}) \leq \theta, \tau(P_{jd}) < T - \theta\}. \tag{14}$$

Now, the dynamic flow Φ is defined by

$$\Phi_e(\theta) = \sum_{P \in \mathcal{P}_e(\theta)} x_P \quad \forall e \in A, \theta \in [0, T]. \quad (15)$$

2.2.2. *Discrete Dynamic Flow.* Discretizing the time intervals $[0, T]$ into the time steps $0, 1, \dots, T - 1$, each corresponding to $[0, 1), [1, 2), \dots, [T - 1, T)$, we can replace the integral sign in (9) by a summation sign (removing $d\sigma$); the corresponding flow is known as discrete dynamic flow. Using the concept of natural transformations, Fleischer and Tardos [16] show the equivalence between the two problems so that the solution procedures of a problem in continuous time version can be carried to the solution procedure of the corresponding problem in the discrete version, and vice versa.

3. Dynamic Contraflow Solutions

We consider the network N with set of nodes V , set of arcs A , capacity $u: A \rightarrow \mathbb{R}_{\geq 0}$, and travel time functions $\tau, \bar{\tau}: A \rightarrow \mathbb{R}_{\geq 0}$. For each $e \in A$, with the tail node i and head node j , $\tau(e)$ or τ_e denotes the arc transit time from i to j and $\bar{\tau}(e)$ or $\bar{\tau}_e$ denotes arc transit time from j to i . Without loss of generality, we make the following conventions:

- (1) There exist at most two arcs (with opposite orientations) between any two nodes i and j . We denote the arc with the tail i and the head j by (i, j) .
- (2) Whenever $(i, j) \in A$, there is $(j, i) \in A$. This can be done, for our purpose, by assigning $u_{ji} = 0$ if such an arc does not exist.
- (3) Defining $\tau(j, i) = \bar{\tau}(i, j)$, we assume the existence of only one travel time function τ .

Remark 1. In case the above-mentioned conventions are not satisfied, we can use suitable network transformations to meet the requirements. For example, if $\tau(i, j) \neq \bar{\tau}(i, j)$, we can transform the network N_0 in Figure 2(a) to the network N shown in Figure 2(b). Given the capacities $u(i, j) = u_{ij}, u(j, i) = u_{ji}$ and transit times $\tau(i, j) = \alpha_1, \bar{\tau}(i, j) = \alpha_2, \tau(j, i) = \beta_1, \bar{\tau}(j, i) = \beta_2$, we replace arc (i, j) in N_0 by two arcs $(i, j), (j, i)$ with capacities and transit times u_{ij}, α_1 and $0, \alpha_2$, respectively. The arc (j, i) in N_0 is replaced by $(i, k), (k, j), (j, k), (k, i)$ adding a node k to the network to avoid parallel arcs. The capacities and travel times of $(j, k), (k, i)$ are taken as $u_{ji}, 0$ and u_{ji}, β_1 , respectively, and those of $(i, k), (k, j)$ are taken as $0, \beta_2$, and $0, 0$, respectively.

Definition 1 (Auxiliary network). For each $N = (V, A, u, \tau, s, d)$, we define the auxiliary network as $N' = (V', A', u', \tau', s, d)$, in which

- (1) $V' = V, A' = A$
- (2) $\forall (i, j) \in A, u'(i, j) = u(i, j) + u(j, i)$
- (3) $\forall (i, j) \in A', \tau'(i, j) = \tau(i, j)$

Example 1. Consider a network N as depicted in Figure 3(a). The arc labels represent the capacity and the transit time. The

auxiliary network N' of N is constructed in Figure 3(b). The capacity of each arc is the sum of its capacity and the opposite arc and the transit time is the same as that of the corresponding arc in N .

3.1. Maximum Dynamic Contraflow

Problem 1 (maximum dynamic contraflow problem with orientation-dependent transit times). Given a network $N = (V, A, u, \tau, s, d)$ with transit time τ depending on the orientation and a time horizon T , find the maximum dynamic flow allowing the arc reversals at time 0.

According to Ford and Fulkerson [17], the problem of finding the static flow corresponding to the temporally repeated maximum dynamic flow can be formulated as

$$\begin{aligned} \min -Tv + \sum_{(i,j) \in A} \tau(i,j)x(i,j), \\ \sum_{e \in A_i^+} x(i,j) - \sum_{(i,j) \in A_i^-} x(i,j) = \begin{cases} v, & i = s \\ -v, & i = d \\ 0, & i \in V \setminus \{s, d\} \end{cases}, \\ 0 \leq x_{(i,j)} \leq u(i,j) \forall (i,j) \in A. \end{aligned} \quad (16)$$

Problem 1 is to find the maximum dynamic flow so that an arc (i, j) can take also the capacity of (j, i) and vice versa. So, $x(i, j) + x(j, i)$ must not exceed $u(i, j) + u(j, i)$. However, the removal of cycle flows does not change the value of the static flow v and does not improve $-Tv + \sum_{(i,j) \in A} \tau(i,j)x(i,j)$; we can impose the condition that either $x(i, j) = 0$ or $x(j, i) = 0$. So, the problem to find the static flow corresponding temporally repeated maximum dynamic flow can be stated as

$$\begin{aligned} \min -Tv + \sum_{(i,j) \in A} \tau(i,j)x(i,j), \quad (17) \\ \sum_{e \in A_i^+} x(i,j) - \sum_{(i,j) \in A_i^-} x(i,j) = \begin{cases} v, & i = s, \\ -v, & i = d, \\ 0, & i \in V \setminus \{s, d\}, \end{cases} \quad (18) \\ 0 \leq x_{(i,j)} \leq u(i,j) + u(j,i) \forall (i,j) \in A, \quad (19) \\ x(i,j) \cdot x(j,i) = 0. \quad (20) \end{aligned}$$

See also [17] for the similar formulation of maximizing the static flow in undirected and mixed networks. The problem in (17)–(19) is a linear programming problem and (20) can be satisfied by the removal of cycle flows in the solution. Since a linear programming problem is polynomial solvable and using flow decomposition, removal of cycle flows also can be done in polynomial time (see [14]); we can find the static flow corresponding to the temporally repeated maximum dynamic flow allowing arc reversals in a

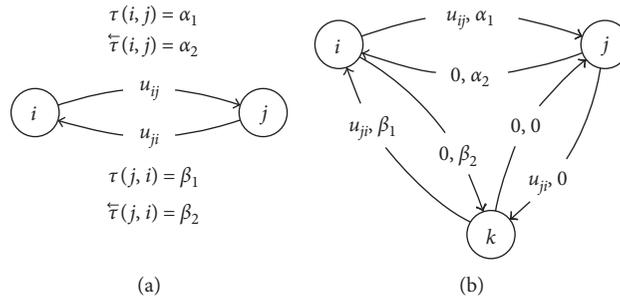


FIGURE 2: Network transformation for the cases $\tau(j, i) \neq \bar{\tau}(i, j)$. (a) Network N_0 . (b) Network N .

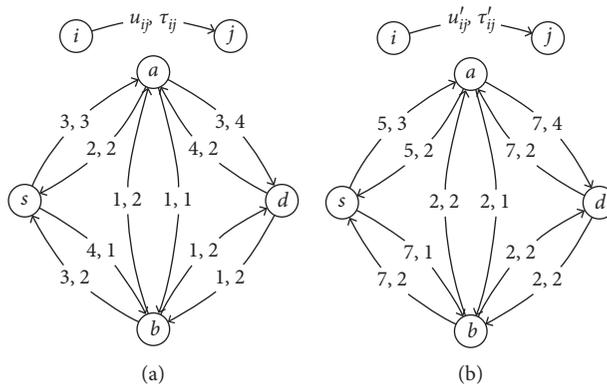


FIGURE 3: (a) Evacuation network N . (b) Auxiliary network N' .

polynomial time. If the static flow in an arc exceeds its capacity, the opposite arc has to be reversed at time zero.

In Algorithm 1, we present a procedure to solve Problem 1.

Theorem 1. Algorithm 1 solves the maximum dynamic contraflow problem with orientation-dependent transit times correctly.

Proof. It is easy to see that steps 1–3 are well defined. Step 4 may be ill defined if for some $(i, j) \in A'$, $x(i, j), x(j, i) > 0$. But because the removal of cycle flows in Step 3 ensures either $x(i, j)$ or $x(j, i)$ to be zero, Step 4 is also well defined. Moreover, $x(i, j) \leq u'(i, j) = u(i, j) + u(j, i)$. This shows that x is feasible in N after necessary arc reversals.

Since the cycle flows do not contribute to the value of the flow, x is optimal in N' even after removing the cycle flows in Step 3. Let v^x be the value of such a dynamic flow. We claim that x is optimal in N (after arc reversals). If not so, there exists an instance of arc reversals, N^y , of N in which we can find a static flow y , temporal repetition of which results in a dynamic flow with flow value more than v^x . In N^y , we can replace (i, j) and (j, i) by a single arc (i, j) with transit time $\tau(i, j)$ and capacity $u(i, j) + u(j, i)$ if (i, j) has been reversed. Let \mathcal{P} be the set of paths corresponding to the path decomposition of y . Corresponding to each path $P: s-i_1-i_2-\dots-i_k-d$ in \mathcal{P} , we have a path $P': s-i_1-i_2-\dots-i_k-d$ in N' . Each arc in P' has the same transit time and the capacity not less than that of the corresponding arc in P . Let the

collection of such paths P' be \mathcal{P}' . Defining a flow $y'(i, j) = y(i, j)$ for each $(i, j) \in \mathcal{P}'$, we can find a dynamic flow in N' with a flow value more than v^x . This contradicts the optimality of x in N' also.

Hence, Algorithm 1 computes the maximum dynamic flow in N reversing appropriate arcs. \square

Theorem 2. Algorithm 1 runs in $O(MDF + nm)$ time where $n = |V|, m = |A|$ and MDF is the running time of a temporally repeated maximum dynamic flow computation.

Proof. The construction of the auxiliary network in Step 1 requires $O(m)$ time. The flow decomposition in Step 3 can be done in $O(mn)$ time (Ahuja et al.) [14]. Step 4 can be performed in $O(m)$ time. Thus, if MDF is the running time of Step 2, the algorithm runs in $O(MDF + nm)$ time. However, in general, this complexity could be considered as the complexity of MDF solution. \square

Theorem 3. The maximum dynamic contraflow problem with orientation-dependent transit times can be solved in a strongly polynomial time.

Proof. From Theorem 2, the running time of Algorithm 1 is $O(MDF + nm)$, where MDF is the running time of a maximum dynamic flow calculation. In $N = (V, A, u, \tau, s, d)$, and consequently in the auxiliary network N' , we can find the static flow x corresponding to the maximum dynamic flow as follows [16]:

Input: A network $N = (V, A, u, \tau, s, d)$ with orientation dependent transit time τ , and a time horizon T

Output: Maximum dynamic flow with arc reversals

- (1) Construct the auxiliary network $N' = (V', A', u', \tau', s, d)$ according to Definition 1.
- (2) Find a static flow x corresponding to the temporally repeated dynamic flow in N' .
- (3) Decompose x into chain-flows and cycle-flows. Remove the cycle-flows and update x .
- (4) Reverse $(j, i) \in A$ if and only if $x(i, j) > u(i, j)$. Dynamic flow in reconfigured $N =$ a temporally repeated flow corresponding to x with the time horizon T .

ALGORITHM 1: Maximum dynamic contraflow with orientation-dependent transit times.

- (1) Add an arc (d, s) with $u(d, s) = \infty$ and $\tau'(d, s) = -T$ to the auxiliary network N' to obtain a network N'' .
- (2) Calculate the minimum cost circulation x' in N'' .
- (3) $x =$ the restriction of x' to N' .

Using the enhanced capacity scaling algorithm mentioned by Orlin [18], the minimum cost circulation in N'' can be calculated in $O(m \log n(m + n \log n))$ time. This proves the assertion. \square

3.2. Quickest Contraflow

Problem 2. (Quickest contraflow problem with orientation-dependent transit times). Given a network $N = (V, A, u, \tau, s, d)$ with transit time τ depending on the orientation and a supply Q at s , find the dynamic flow with minimum time horizon allowing the arc reversals at time 0.

We construct Algorithm 2 to solve Problem 2.

Let $v(T)$ be the value of the maximum dynamic flow with time horizon T . One of the strategies to find the quickest flow with a supply Q at s is to search for the minimum time horizon T^* such that $Q \leq v(T^*)$. Various search strategies are described in Burkard et al.'s work [19]. This requires solving the maximum dynamic flow problem repeatedly. Using this technique in Step 2, the feasibility and optimality arguments given in Theorem 1 are valid in Algorithm 2 as well. Hence, we have the following result.

Theorem 4. *Algorithm 2 solves the quickest contraflow problem with orientation-dependent transit times correctly.*

Analogous to the case of solving the maximum dynamic contraflow problem, if QF is the complexity of solving the quickest flow problem in N , we have the following.

Theorem 5. *Algorithm 2 runs in $O(QF + nm)$ time, where QF is the running time of a temporally repeated quickest flow problem.*

Since there exist strongly polynomial algorithms for solving the quickest flow problem, we have the following.

Theorem 6. *The quickest contraflow problem with orientation-dependent transit times can be solved in a strongly polynomial time.*

Proof. Consider a network $N = (V, A, u, \tau, s, d)$ with a supply Q at the source s . We show that the complexity of Algorithm 2 is strongly polynomial to reach the conclusion. Using the cancel-and-tighten algorithm described in the work of Saho and Shigeno [20], to solve the quickest flow problem Step 2 takes $O(nm^2 \log^2 n)$ time. Using the fact in Theorem 5, the complexity of Algorithm 2 is $O(nm^2 \log^2 n)$. This proves the assertion. \square

3.3. Earliest Arrival Contraflow. Consider a network $N = (V, A, u, \tau, s, d)$ with orientation-dependent transit time τ . With a given time horizon T , the earliest arrival flow problem seeks to find the flow that is maximum with each time horizon $\theta \in [0, T]$. Carrying over the idea of solving the maximum dynamic contraflow problem, we try to solve the earliest arrival flow problem on the auxiliary network N' . This can be carried out by implementing the successive shortest path algorithm in the residual network of N' ([21, 22]). The idea is to send the flow successively through shortest paths amounting the residual capacity of the path. Such a path, being in the residual network, may contain negative (cost) time arcs also. Sending flow through negative cost arcs sends the flow back in time. This may result in the arc used by the flow in some time interval, not used in the other time interval, and used again in some further time interval. So, an arc reversed to optimize the flow at one time does not remain reversed throughout the given whole-time interval. In other words, an arc may have to be flipped back and forth unlike in the cases of maximum dynamic contraflow and quickest contraflow, where an arc reversed at the beginning (time zero) remains reversed throughout.

Problem 3. Given a network $N = (V, A, u, \tau, s, d)$ with orientation-dependent transit time τ and a time horizon T , find a dynamic flow which is maximum at every time horizon θ such that $0 \leq \theta \leq T$ allowing arc reversals at time θ .

We propose Algorithm 3, which sends the flow along the successive shortest paths in the residual network of the auxiliary network.

The earliest arrival flow in Step 2 can be found by sending the flow equal to the residual capacity along the shortest paths in the residual network of N' . The generalized temporally repeated flow can be found in [15]. Such an algorithm runs in a pseudo-polynomial-time.

Input: A network $N = (V, A, u, \tau, s, d)$ with orientation dependent transit time τ , and a supply Q at s

Output: Quickest flow with the arc reversals

- (1) Construct the auxiliary network $N' = (V', A', u', \tau', s, d)$ according to Definition 1.
- (2) Find a static flow x corresponding to the temporally repeated quickest flow in N' , and the quickest time T^* .
- (3) Decompose x into chain-flows and cycle-flows. Remove the cycle-flows and update x .
- (4) Reverse $(j, i) \in A$ if and only if $x(i, j) > u(i, j)$. Quickest flow in reconfigured N is a temporally repeated flow corresponding to x with the time horizon T^* .

ALGORITHM 2: Quickest contraflow with orientation-dependent transit times.

Input: A network $N = (V, A, u, \tau, s, d)$ with orientation dependent transit time τ , and a time horizon T

Output: Earliest arrival flow with the arc reversals

- (1) Construct the auxiliary network $N' = (V', A', u', \tau', s, d)$ according to Definition 1.
- (2) Find the earliest arrival flow f in N' .
- (3) Reverse $(j, i) \in A$ if $f_{ij}(\theta) > u(i, j)$ at time $\theta \in [0, T]$. f is the earliest arrival flow with arc reversals in N .

ALGORITHM 3: Earliest arrival contraflow with orientation-dependent transit times.

Theorem 7. *Algorithm 3 solves the earliest arrival contraflow problem with orientation-dependent transit times in a pseudo-polynomial-time.*

However, in a series-parallel network with a single source and single sink, the maximum dynamic flow has the earliest arrival property (Ruzika et al.) [23]. So, in case of a series-parallel network N , Algorithm 1 solves Problem 3. Moreover, the maximum dynamic flow in the auxiliary network of a series parallel network with time horizon T can be solved in $O(mn + m \log m)$ time by sending the flow iteratively through the s - d paths with the minimum time and removing the saturated arcs, considering only the paths with time not exceeding T . So, we have the following.

Theorem 8. *If $N = (V, A, u, \tau, s, d)$ is a series-parallel network, then the earliest arrival contraflow problem with orientation-dependent transit times can be solved in $O(mn + m \log m)$ time.*

4. Partial Contraflow Algorithm

The approach described in Section 3 either reverses an arc or does not reverse it. In various applications, for example, evacuation planning, an arc refers to a collection of lanes in a particular direction; it is beneficial if we reverse the lanes required by the flow. The unused lanes may be used for other facilities [24]. Realizing the need of arc reversals up to the required capacity only, the authors in [4] introduce the concept of partial arc reversals. We extend the procedure in case of orientation-dependent transit times on arcs.

Algorithm 4 describes the generic procedure to solve the corresponding problems described in Section 3. The algorithm not only reverses the arcs up to the necessary capacity but also lists the unused capacities of the arcs of the network considered.

The correctness of Algorithm 4 can easily be realized from the correctness of the corresponding algorithm with full arc reversal and the following fact. For the arcs $(i, j), (j, i)$ between nodes i and j , it is evident that either only one of them is reversed or both of them are not reversed. If (i, j) is reversed, it clearly indicates that $x(j, i) > u(j, i)$; there is no capacity of (j, i) unused; that is, $r(j, i) = 0$, and

$$\begin{aligned} r(i, j) &= u(i, j) - [x(j, i) - u(j, i)] \\ &= u(i, j) + u(j, i) - x(j, i) \\ &= u'(i, j) - x(j, i). \end{aligned} \quad (21)$$

If both arcs are not reversed, then $x_{ij} \leq u(i, j)$ meaning $r(i, j) = u(i, j) - x(i, j)$.

The flow x in Step 1 of Algorithm 4 has to be considered as the corresponding static flow in case of the maximum dynamic contraflow (Section 3.1) and quickest flow (Section 3.2) and the dynamic flow in case of the earliest arrival flow (Section 3.3). Step 2 has to be implemented at time zero and at time θ accordingly.

As the extra procedure of listing the unused capacities in Step 3 takes only $O(m)$ time, the overall complexities of the algorithms in case of partial contraflow are the same as those of the contraflow. Thus, we have the following.

Theorem 9. *The worst-case complexity of a partial contraflow problem with orientation-dependent transit times is the same as that of the corresponding contraflow problem.*

5. Computational Experiment

For testing the computational performance of the maximum dynamic contraflow algorithm (Algorithm 1) and the quickest contraflow algorithm (Algorithm 2), we consider Kathmandu road network inside Ring Road with major road segments as an example network (Figure 4). We consider a

Input: Network $N = (V, A, u, \tau, s, d)$ with orientation dependent transit times
Output: Partial contraflow reconfiguration of N with unused capacities r .
 (1) Find a cycle-free flow x in the auxiliary network $N' = (V', A', u', \tau', s, t)$.
 (2) Reverse $(j, i) \in A$ proportional to the capacity $x(i, j) - u(i, j)$ if $x(i, j) > u(i, j)$.
 (3) For each $(i, j) \in E$, if (i, j) is reversed, then $r_{ij} = u'_{ij} - x_{ji}$ and $r_{ji} = 0$. If neither (i, j) nor (j, i) is reversed, $r(i, j) = u(i, j) - x(i, j)$.

ALGORITHM 4: Generic partial contraflow algorithm.

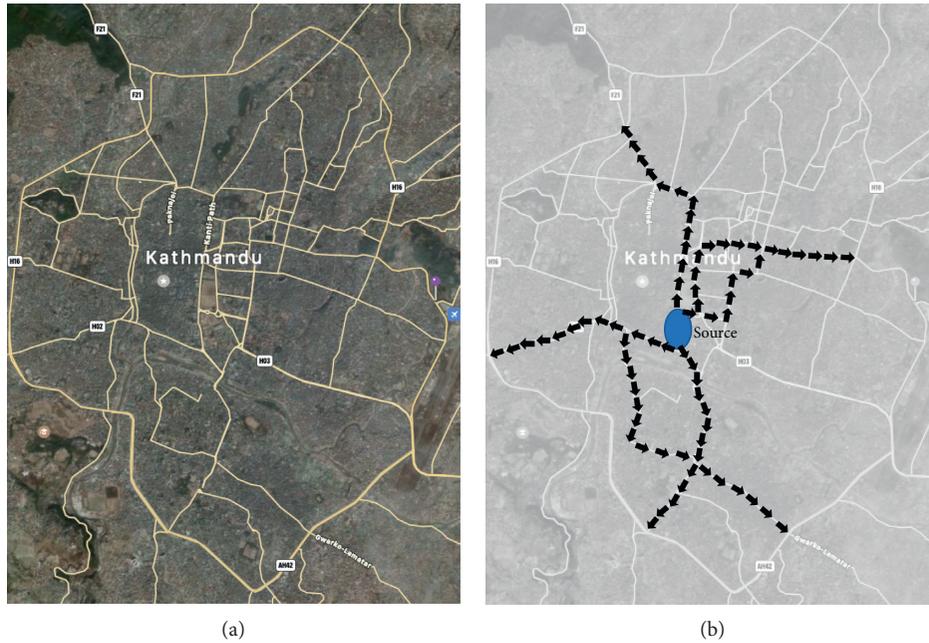


FIGURE 4: (a) Kathmandu road network. (b) Maximum flow direction with arc reversals $T = 1$ hour.

TABLE 1: Network data for computations (Section 5).

i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}	i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}	i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}
0	49	3	3	85	70	0	1	4	0	70	70	0	16	0	4	36	36
0	19	2	2	41	41	1	2	4	0	18	16	1	14	2	2	18	18
2	3	2	2	47	47	2	12	4	0	21	21	3	4	2	2	32	25
3	11	2	2	21	21	4	5	2	2	10	25	4	56	2	2	35	50
5	6	2	2	180	150	5	9	2	2	65	70	6	7	2	2	90	110
7	8	2	2	85	95	7	33	2	2	45	55	8	9	2	2	30	25
8	30	2	2	25	30	8	32	2	2	24	30	9	10	2	2	35	45
10	11	2	2	34	34	10	27	2	2	30	35	10	30	2	2	64	64
11	12	2	2	30	35	12	13	4	0	20	25	12	27	2	2	40	40
13	14	4	0	26	26	14	15	4	0	40	42	14	25	2	2	55	60
15	16	4	0	24	27	15	24	2	2	50	50	16	22	0	4	52	52
16	23	2	2	41	41	19	20	2	2	50	50	19	22	2	2	45	40
20	21	2	2	100	105	20	48	2	2	130	145	21	47	2	2	50	50
21	48	2	2	153	153	22	23	2	2	36	40	22	45	4	4	145	145
23	24	2	2	34	30	24	25	2	2	42	38	25	26	2	2	30	25
25	42	0	3	36	28	26	27	2	2	18	16	27	28	2	2	32	32
28	29	2	2	34	30	28	41	2	2	17	15	29	30	2	2	18	16
29	40	2	2	24	24	30	31	2	2	24	28	31	32	2	2	25	22
32	33	2	2	22	20	33	34	2	2	20	25	34	35	2	2	32	30

TABLE 1: Continued.

i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}	i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}	i	j	u_{ij}	u_{ji}	τ_{ij}	τ_{ji}
34	37	2	2	88	80	35	37	2	2	24	28	36	37	2	2	35	40
38	39	2	2	65	75	39	40	2	2	47	47	39	43	2	2	32	35
40	41	2	2	30	34	41	42	2	2	30	34	41	43	2	2	37	37
42	44	2	2	120	135	44	45	2	2	135	135	45	46	2	2	135	135
48	49	2	2	126	126	49	50	2	2	38	38	50	51	2	2	38	38
50	64	2	2	20	20	51	52	2	2	80	75	51	63	2	2	130	125
51	64	2	2	45	45	52	53	2	2	50	45	52	61	2	2	40	35
53	54	2	2	18	18	53	61	2	2	25	22	54	55	2	2	68	68
54	58	2	2	70	70	55	56	2	2	36	34	55	57	2	2	76	70
55	58	2	2	82	75	57	58	2	2	80	75	58	59	2	2	45	40
59	60	2	2	60	65	60	62	2	2	34	34	62	63	2	2	68	68
64	101	2	2	180	160	101	999	2	2	0	0	63	999	2	2	60	60
62	999	2	2	108	100	60	999	2	2	117	110	59	104	2	2	47	47
104	999	2	2	0	0	59	999	2	2	45	45	57	999	2	2	21	21
56	999	2	2	171	171	6	999	2	2	108	108	35	999	2	2	153	153
36	110	2	2	90	100	110	999	2	2	0	0	36	999	2	2	70	65
38	112	2	2	50	40	112	999	2	2	0	0	38	999	2	2	41	36
43	999	2	2	100	90	44	999	2	2	145	135	46	999	4	4	50	50
21	999	2	2	190	200	47	118	2	2	198	198	118	999	2	2	0	0
47	999	2	2	80	80	64	999	2	2	135	135						

Source node: 0; sink node: 999.

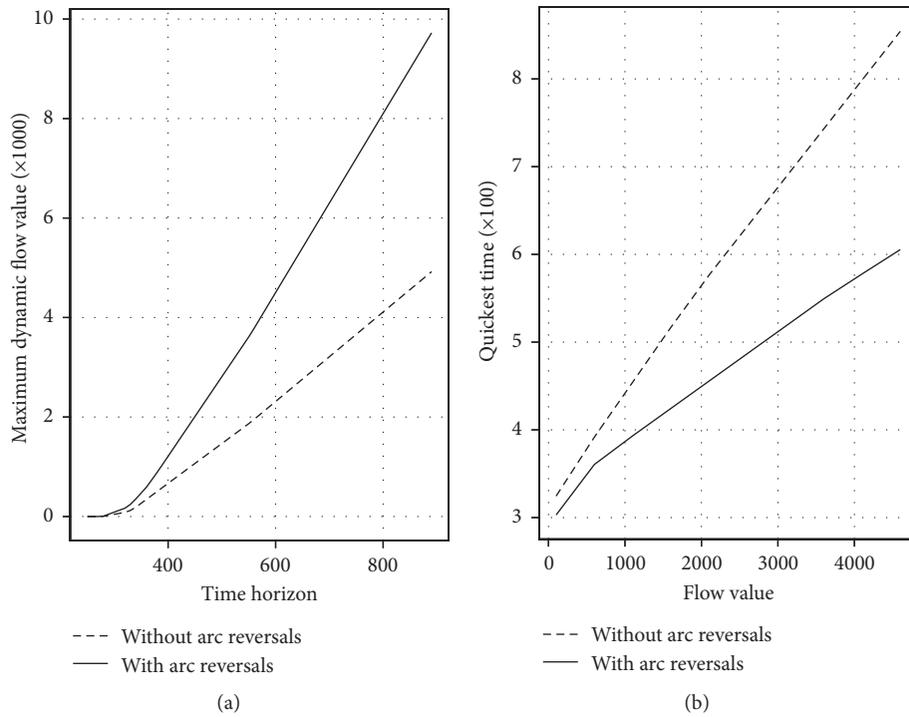


FIGURE 5: Flow value and time comparisons.

problem of evacuating people in Dasharath Stadium and surrounding area to outside of the Ring Road in case of auto-based-emergency evacuation.

The constructed directed network consists of 69 nodes and 232 arcs (see Table 1). The direction of arcs is taken as the direction of the usual flow of the traffic in the

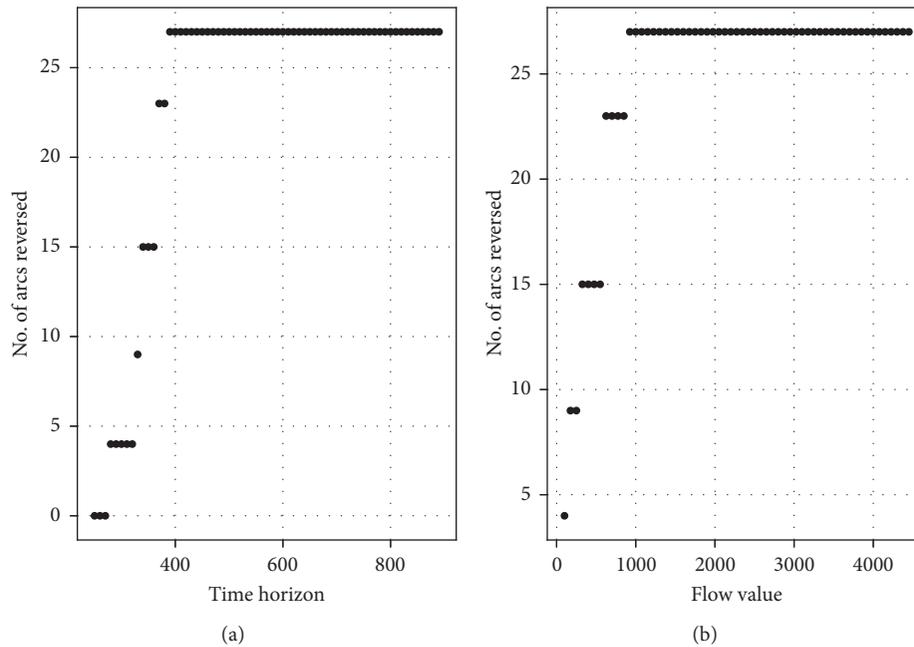


FIGURE 6: Number of arcs reversed.

corresponding segments. The capacities of the road segments are taken from 2 to 4 units of flow per second according to the number of lanes. The travel time (in seconds) in one of the arcs between any two nodes is considered according to the length and that in the opposite arc is chosen differing from it by 0 to 30 seconds (chosen randomly). The considered data are taken only for the purpose of testing the algorithms. The accuracy of capacity and travel time demands complex technical examinations.

We calculate the maximum dynamic flow with and without contraflow taking time horizons from as low as 5 minutes to as high as 1 hour. At each time horizon considered, we find that the value of the flow after allowing arc reversals is almost the double of that without arc reversals (Figure 5(a)). At $T = 5$ minutes, it is 44 without arc reversals and 88 with arc reversals. At $T = 60$ minutes, the corresponding values are 29,312 and 58,502.

Given a flow value at the source, the calculation of quickest flow shows that the decrease in the quickest time increases with the increase in the flow value. With the flow value as low as 500, the quickest time decreases only by 7%, which is 47% for the flow value 50,000.

With growing time horizon and growing flow value, as well as the number of arcs, the flow occupies more and more arcs and, consequently, the number of arcs reversed increases. However, it remains fixed after some value of time or the flow value (Figure 6(a) for maximum dynamic flows calculations and Figure 6(b) for quickest flows).

Among the considered instances, the running time of a maximum contraflow calculation is at most 0.013 seconds, and that of a quickest contraflow calculation is at most 0.067 seconds. The coding is done in Python programming language and run in MacOS 11.1 with 1.8 GHz Dual-Core Intel Core i5 processor, and 8 GB RAM.

6. Conclusion

In this work, we introduce the contraflow problem in which the transit time on an arc depends on the direction of the arc; that is, the transit time on an arc may change after its reversal. This extends the notion, in the existing literature, that the transit time on arcs remains the same before and after the arc reversal to the cases where the time on arcs depends on its orientation. Presenting a method of constructing an auxiliary network, strongly polynomial time algorithms for maximum dynamic contraflow problem and quickest contraflow problem with orientation-dependent transit times are presented for a single-source-single-sink network. In the similar settings, for the earliest arrival contraflow problem, a pseudo-polynomial-time algorithm is also presented. The computational performance of the algorithms for maximum dynamic contraflow and quickest contraflow taking a Kathmandu road network is also tested.

The presented approach is useful, particularly, in transportation planning, where the transit time depends on the direction of the traffic flow because of various reasons, for example, topography of the road. When the direction of the traffic flow in a road segment is reversed, if the capacity permits, it is beneficial to reverse only the necessary lanes. To address such an issue, we present corresponding algorithms in the partial contraflow setting as well. Analyzing impressive results from this research, its further extensions to flow-dependent scenarios would be interesting problems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The first author acknowledges the support of UGC Nepal for Ph.D. fellowship, and the second author acknowledges the support of AvH Foundation for the return fellowship.

References

- [1] T. N. Dhamala, U. Pyakurel, and S. Dempe, "A critical survey on the network optimization algorithms for evacuation planning problems," *International Journal of Operations Research*, vol. 15, pp. 101–133, 2018.
- [2] I. S. Kotsireas, A. Nagurney, and P. M. Pardalos, *Dynamics of Disasters-Algorithmic Approaches and Applications*, Springer Optimization and Its Applications, New York, NY, USA, 2018.
- [3] S. Rebennack, A. Arulselvan, L. Elefteriadou, and P. M. Pardalos, "Complexity analysis for maximum flow problems with arc reversals," *Journal of Combinatorial Optimization*, vol. 19, no. 2, pp. 200–216, 2010.
- [4] U. Pyakurel, H. N. Nath, S. Dempe, and T. N. Dhamala, "Efficient dynamic flow algorithms for evacuation planning problems with partial lane reversal," *Mathematics*, vol. 7, pp. 1–29, 2019.
- [5] P. P. Bhandari and S. R. Khadka, "Evacuation contraflow problems with not necessarily equal transit time on anti-parallel arcs," *American Journal of Applied Mathematics*, vol. 8, pp. 230–235, 2020.
- [6] U. Pyakurel and T. N. Dhamala, "Models and algorithms on contraflow evacuation planning network problems," *International Journal of Operations Research*, vol. 12, pp. 36–46, 2015.
- [7] U. Pyakurel and T. N. Dhamala, "Continuous time dynamic contraflow models and algorithms," *Advances in Operations Research*, vol. 2016, Article ID 368587, 7 pages, 2016.
- [8] S. Kim, S. Shekhar, and M. Min, "Contraflow transportation network reconfiguration for evacuation route planning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 1–15, 2008.
- [9] U. Pyakurel, T. N. Dhamala, and S. Dempe, "Efficient continuous contraflow algorithms for evacuation planning problems," *Annals of Operations Research*, vol. 254, no. 1-2, pp. 335–364, 2017.
- [10] U. Pyakurel, H. N. Nath, and T. N. Dhamala, "Partial contraflow with path reversals for evacuation planning," *Annals of Operations Research*, vol. 283, no. 1-2, pp. 591–612, 2019.
- [11] U. Pyakurel, H. N. Nath, and T. N. Dhamala, "Efficient contraflow algorithms for quickest evacuation planning," *Science China Mathematics*, vol. 61, no. 11, pp. 2079–2100, 2018.
- [12] R. C. Dhungana and T. N. Dhamala, "Flow improvement in evacuation planning with budget constrained switching costs," *International Journal of Mathematics and Mathematical Sciences*, vol. 2020, Article ID 1605806, 10 pages, 2020.
- [13] U. Pyakurel and S. Dempe, "Network flow with intermediate storage: models and algorithms," *SN Operations Research Forum*, vol. 1, pp. 1–23, 2020.
- [14] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, NJ, USA, 1993.
- [15] M. Skutella, "An introduction to network flows over time," in *Research Trends in Combinatorial Optimization*, pp. 451–482, Springer, Berlin, Germany, 2009.
- [16] L. Fleischer and É. Tardos, "Efficient continuous-time dynamic network flow algorithms," *Operations Research Letters*, vol. 23, no. 3-5, pp. 71–80, 1998.
- [17] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 1962.
- [18] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, vol. 41, no. 2, pp. 338–350, 1993.
- [19] R. E. Burkard, K. Dlaska, and B. Klinz, "The quickest flow problem," *ZOR Zeitschrift für Operations Research Methods and Models of Operations Research*, vol. 37, no. 1, pp. 31–58, 1993.
- [20] M. Saho and M. Shigeno, "Cancel-and-tighten algorithm for quickest flow problems," *Networks*, vol. 69, no. 2, pp. 179–188, 2017.
- [21] E. Minieka, "Maximal, lexicographic, and dynamic network flows," *Operations Research*, vol. 21, no. 2, pp. 517–527, 1973.
- [22] W. L. Wilkinson, "An algorithm for universal maximal dynamic flows in a network," *Operations Research*, vol. 19, no. 7, pp. 1602–1612, 1971.
- [23] S. Ruzika, H. Sperber, and M. Steiner, "Earliest arrival flows on series-parallel graphs," *Networks*, vol. 57, no. 2, pp. 169–173, 2011.
- [24] H. N. Nath, U. Pyakurel, T. N. Dhamala, and S. Dempe, "Dynamic network flow location models and algorithms for evacuation planning," *Journal of Industrial and Management Optimization*, vol. 13, 2020.