IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 7 Issue 7, July 2020
**ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72**
www.ijiset.com

# Efficient Lane Reversals for Prioritized Maximum Flow

**Urmila Pyakurel[1*], Sachin Wagle[2] and Mohan Chandra Adhikari[3]**

[1]Associate Professor, Central Department of Mathematics, IOST, Tribhuvan University, Kathmandu, 44600, Nepal

[2]MPhil Scholar, Central Department of Mathematics, IOST, Tribhuvan University, Kathmandu, 44600, Nepal

[3]MPhil Scholar, Central Department of Mathematics, IOST, Tribhuvan University, Kathmandu, 44600, Nepal

[*] Corresponding author: Urmila Pyakurel, Central Department of Mathematics, IOST, Tribhuvan University, Kathmandu, 44600, Nepal

**Abstract**

The model of lane reversals in the network flow theory is broadly accepted world-wide to increase the flow value by increasing the outbound arc capacity from the sources towards the sinks with lane reversals capability. There are different approaches, e.g., simulation, heuristic and analytical that are dealing this issue extensively. Specially, this model is used to manage the empty road during evacuation planning so that maximum number of evacuees can be shifted from accidental areas to the safe areas as quickly and efficiently as possible.

In this paper, we focus on the analytical approach of lane reversals in which the intermediate storage is allowed. We consider a network with multiple sources and multiple sinks having given priority ordering of each source and sink. We introduce the prioritized maximum flow problem with intermediate storage allowing lane reversals and present polynomial time algorithms to solve them in both static and dynamic networks.

***Keywords:*** *Network flow, lane reversals, intermediate storage, priority ordering, maximum flow*

## 1. Introduction

The application of network flow theory in the evacuation planning is highly investigated in the literature, [4, 13]. The evacuation scenario is represented by a network in which the accidental areas are sources, safe areas are sinks, the connections between them are lanes or arcs, the intersection of the arcs are the intermediate nodes. Each node has finite capacity and each arc has capacity (an upper bound of the flow), transit time (times required to travel the arc) or cost needed to transship unit flow along the arcs. A maximum flow provides the largest possible amount of flow from the sources to the sinks in a given network within a given time horizon $T$. The maximum flow problem was discussed as a simplified model of railway traffic flow in Russia, [8]. Ford and Fulkerson [6 ,7] introduced the maximum dynamic flow problem and developed the first well known algorithm that sends maximum flow from the source to the sink by augmenting along *s-d* paths and proved the maximum amount of flow is equal to the total capacity of the arcs in the minimum cut. The prioritized maximum flow problem was introduced on the multi-terminal static network and solved it in polynomial time, [14]. The maximum flow at every time point, i.e., the earliest arrival flow problem has been solved in pseudo-polynomial time complexity on two terminal network in [14]. By reducing the network into two terminal series parallel, this problem has been solved in strongly polynomial time, [28]. For the dynamic network, the prioritized maximum flow problem has been solved with a polynomial time algorithm, [9, 10].

Recently, the network flow models with intermediate storage, i.e., the inflow into intermediate nodes is allowed to be greater than the outflow, and the excess flow can be stored in that node provided it does not exceed the node capacity, has been introduced and presented polynomial time algorithms to solve the maximum static, maximum dynamic, earliest arrival flow problems in [17, 18]. Their main aim is to use maximum arc capacity to push as much flow as possible out of the sources so that their model can only be used if the total capacity of arcs leaving the source is not smaller than the minimum cut capacity of the network. Moreover, the excess flow is sent as far as possible from the sources. Similarly, the prioritized maximum flow with intermediate storage has been computed in both static and dynamic networks using polynomial time algorithms in [16]. The non-conservative aspect of the network flow problem has been considered, however, a theoretical validity proof and an experimental verification are lacking, [11].

The lane reversals technique is developed to solve the evacuation planning problem. The evacuation planning problem is the process of shifting residents from disastrous areas to safe places as quickly and efficiently as possible. After disasters, the lanes outwards from the sources are congested due to the large number of evacuees on the street but the movement of flow towards the sources is not allowed so that these lanes remain empty. The concept of lane reversals has been introduced using graph and flow theory for the evacuation planning problem, [12]. They proved that the problem on multiple sources and sinks is NP-complete and presented two heuristic algorithms; greedy and bottleneck relief to solve it. For the large size problem a heuristic algorithm, reversing at most a given number of arcs, has been presented and solved it efficiently, [29]. Their main objective is to minimize the number of vehicles that have to spend time on the most endangered nodes. The investigation of the large size lane management problem with discrete time traffic assignment models to minimize the overall delay can be found in [30]. By introducing penalty function on each arc with capacity and transit time, an optimal priority evacuation tree has been presented to minimize the expected evacuation penalty, [1]. With lane reversals and convergence policies, the evacuation planning problem has been investigated on tree network, [3].

The analytical approach, for the maximum flow problem allowing lane reversals has been introduced in both static and dynamic network, and presented polynomial time algorithms to solve them, [2, 27]. By solving the maximum flow at each time point, the earliest arrival flow with lane reversals has been introduced and solved efficiently, [20]. The prioritized maximum flow with lane reversals in both static and dynamic networks have also been investigated and presented polynomial time algorithms to solve both problems, [20]. The technique of lane reversals in continuous time dynamic network has been investigated and presented efficient algorithms to solve a number of dynamic flow problems, in [21, 22, 23]. The quickest flow problem with lane reversals has been investigated and presented first minimum cost flow algorithm to solve it, [24]. By locating the emergency facility at arcs, the problem is solved in polynomial time with lane reversals in [15]. The procedures of partial lane reversals have been introduced and presented a number of efficient algorithms to solve the dynamic network flow problems, [25, 26].

The lane reversal technique for the network flow with intermediate storage has been introduced in which the maximum dynamic and earliest arrival flow problems with intermediate storage allowing lane reversals has been solved with polynomial time algorithms, [17, 18]. However, the possibility of intermediate storage by placing emergency units at the intermediate nodes has been discussed to minimize the evacuation time by allowing lane reversals in [19].

In this paper, we investigate the prioritized maximum flow problem, i.e., flow is maximized with fixed priority ordering of sources and sinks in given time horizon, with intermediate storage by allowing lane reversals. The lexicographic property, i.e., the prioritized maximum flow, of [9, 10] has been used to solve the maximum dynamic flow problem with intermediate storage in both cases allowing lane reversals and without it in [17, 18]. However, we solve the problem independently here. That means, we maximize the flow reaching to the sinks in their priority ordering and push the excess flow as far as possible from the sources with respective priority ordering. In our network, the priority ordering of sources and sinks are given. We present polynomial time algorithms to solve the prioritized maximum static and dynamic flow problems with intermediate storage allowing lane reversals in multi-terminal networks. We adopt the lane reversals technique of [17, 18] to compute the prioritized maximum flow with intermediate storage from [16].

This paper is organized as follows: In Section 2, we describe notations and flow models with intermediate storage allowing lane reversals. The prioritized maximum flow problem with intermediate storage allowing lane reversals is introduced and solved with polynomial time algorithm in Section 3. Similarly, the problem is solved in dynamic network with polynomial time algorithm in Section 4. The paper is concluded in Section 5.

## 2. Preliminaries

Let $S = \{S_x : x = 1,2, \dots, q\}$, $D = \{D_y : y = 1,2, \dots, r\}$ and $I = \{I_z : z = 1,2, \dots, p\}$ be the set of multiple sources, multiple sinks and intermediate vertices, respectively, and $e' = (v, w) \in A$ be an arc with its reverse arc $e^{-1} = (w, v) \in A$, where $A$ is the set of $m$ arcs. The functions $u: A \to \mathbb{Z}^+$, $h: I \to \mathbb{Z}^+$ and $c: A \to \mathbb{Z}^+$ determine the capacity of the arcs, $e', e^{-1} \in A$, the holdover capacity of the nodes $v \in I$ and per unit cost along the arc, respectively. With these parameters, our multiple terminals static network is $N = (V, A, u, h, c)$, where $V = S \cup D \cup I$ is the set of $n$ nodes. For simplicity, we assume that $s$ and $d$ are a single source and a single sink, respectively. If we are given total time period $T$ within which we have to transship the total flow from the sources to the sinks, then the network is dynamic with the parameters $N = (V, A, u, h, \tau,$

$T$), where the function $\tau: A \rightarrow \mathbb{Z}^+$ determines the required time to travel from node $v$ to $w$ in the discrete time $\boldsymbol{T} = \{0, 1, \ldots, T\}$. We also assume that $\tau(e') = \tau(e^{-1})\ \forall e', e^{-1} \in A$.

Our aim is to transship as much flow as possible from the sources to the sinks and to push the excess flow as far as possible from the sources to store at intermediate nodes. One important technique to increase $s - d$ flow in the network is to use the reverse arc capacity as well. As the arcs towards the sources remain empty, they can be used to increase flow by reversing directions. The lane reversal approach has been studied from the emergency point of view. On network $N = (V, A, u, h, c, S, D, I)$, each arc $e'$ and $e^{-1}$ have certain capacity $u(e')$ and $u(e^{-1})$. When we use the lane reversal technique, the arc capacity can be redefined as auxiliary arc capacity $u(e) = u(e') + u(e^{-1})\ \forall e', e^{-1} \in A$ and $e \in E$, where $E$ is the set of auxiliary arcs and $|E| < m$. For the dynamic network $N = (V, A, u, h, \tau, S, D, I,\ T)$, we assume that the transit time $\tau(e) = \tau(e')$ if $e' \in A$, otherwise $\tau(e) = \tau(e^{-1})$. The remaining data are unaltered with lane reversal technique. Moreover, we assume that the intermediate nodes have sufficient capacities and their flow value is bounded by the arc capacities. Thus, we form the auxiliary dynamic network as $N^a = (V, E, u, h, \tau, S, D, I,\ T)$ and auxiliary static network as $N^a = (V, E, u, h, c, S, D, I)$. For any node $v$ the set of incoming and outgoing arcs are denoted as $A_v^- = \{e' \in A: e' = (w, v), w \in V\}$ and $A_v^+ = \{e' \in A: e' = (v, w), w \in V\}$. Generally, we assume $A_S^- = \emptyset = A_D^+$, however, in the case of lane reversals, $A_S^- \neq \emptyset \neq A_D^+$.

According to Ford and Fulkerson [6], maximum $s - d$ flow in a network is equal to the minimum cut capacity $\min u(S, \bar{S}) = \sum_{(v,w)\in(S,\bar{S})} u(v, w)$. If $\sum_{x=1}^{q} f(S_x) > \min \sum_{(v,w)\in(S,\bar{S})} u(v, w)$ then the excess flow does not reach to the sinks. Let us define the static flow of value $val(f)$ by the function $f: A \rightarrow \mathbb{R}^+$ and the amount of flow with value $val(f_v)$ stored at the intermediate nodes by $f_v: I \rightarrow Z^+$. Recently, authors in [17] have addressed the excess flow to hold at the intermediate nodes. Now, the maximum static flow problem with intermediate storage allowing lane reversals is to maximize the objective in Eq. (1) satisfying the conservations in Eq. (2-4).

$$val(f) = \sum_{e\in A_S^+} f(e) = \sum_{e\in A_D^-} f(e) + \sum_{v\in I, h(v)>0} f(v) \tag{1}$$

$$f(v) = \sum_{e\in A_v^-} f(e) - \sum_{e\in A_v^+} f(e) \geq 0,\ \ \forall\ v \in I \tag{2}$$

$$0 \leq f(e) \leq u(e) = u(e') + u(e^{-1})\ \forall\ e',\ e^{-1} \in A\ , e \in E \tag{3}$$

$$0 \leq f(v) \leq h(v)\ \ \forall v \in I \tag{4}$$

The inflow may not be equal to the outflow of each node so that flow conservation constraint is not satisfied in Eq. (2). The excess flow can be stored at the intermediate node. Eq. (3) is the feasibility constraint for each arc in which lane reversals is allowed and flow is bounded by the arc capacity. Similarly, Eq. (4) represents the feasibility condition of node in which the function $h(v)\ \ \forall v \in I$ is sufficiently large and it is bounded below at least the total capacity of arcs incoming to the node.

Let, function $\lambda: A \times \boldsymbol{T} \rightarrow \mathbb{R}^+$ represents a dynamic flow in discrete time $\boldsymbol{T}$ on a dynamic network and $\lambda(v, \theta): I \times \boldsymbol{T} \rightarrow \mathbb{Z}^+$ be the amount of flow stored at the intermediate node $v$ at time $\theta$**.** Then, the maximum dynamic flow problem with intermediate storage allowing lane reversals is to maximize the objective in Eq. (5) satisfying the constraints in Eq. (6-8), as in [17].

$$val(\lambda, T) = \sum_{e\in A_S^+} \sum_{\sigma=0}^{T} \lambda(e, \sigma) = \sum_{e\in A_D^-} \sum_{\sigma=\tau_e}^{T} \lambda(e, \sigma - \tau_e) + \sum_{v\in I, h(v)>0} \lambda(v, T) \tag{5}$$

$$val(\lambda, \theta) = \sum_{e\in a_v^-} \sum_{\sigma=\tau_e}^{\theta} \lambda(e, \sigma - \tau_e) - \sum_{e\in A_v^+} \sum_{\sigma=\tau_e}^{\theta} \lambda(e, \sigma) \geq 0,\ \ \forall v \in I,\ \ \theta \in \boldsymbol{T} \tag{6}$$

$$0 \leq \lambda(e, \theta) \leq u(e, \theta)\ = u(e') + u(e^{-1})\ \forall\ e', e^{-1} \in A\ , e \in E, \forall v \in I, \theta \in \boldsymbol{T} \tag{7}$$

$$0 \leq \lambda(v, \theta) \leq h(v), \qquad \forall v \in I \tag{8}$$

The conservation constraints in Eq. (6) show that the total amount of flow out of the sources may not reach the sinks and the excess is stored at the intermediate vertices in each time period $\theta \in \{0,1,2, \ldots, T\}$. Similarly, the feasibility conditions on the arcs and nodes show that the amount of flow does not exceed their capacity.

## 3. Prioritized maximum static flow with intermediate storage allowing lane reversals

In a network, the maximum $s - d$ flow can be increased by increasing its arcs capacity. One important approach to increase the arcs capacity is to use the arc capacity in the reverse direction. In this section, we investigate the prioritized maximum static flow (PMSF) problem with intermediate storage in reconfigured multiple sources and multiple sinks network and present a polynomial time algorithm to solve it.

**Definition:** If $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_q \subseteq S$, then a maximum flow that shifts $f(S_x)$ units flow from each source $\{S_x: x = 1, 2, \ldots, q\}$is the prioritized maximum flow on the sources. Similarly, if $D_1 \subseteq D_2 \subseteq \cdots \subseteq D_r \subseteq D$, then a maximum flow that delivers $f(D_y)$ units flow into each of the sinks $\{D_y: y = 1, 2, \ldots, r\}$ is the prioritized maximum flow into the sinks, [14]. If $\sum_{x=1}^{q} f(S_x) > \sum_{y=1}^{r} f(D_y)$, then the difference $\sum_{x=1}^{q} f(S_x) - \sum_{y=1}^{r} f(D_y) = \sum_{z=1}^{p} f(I_z)$ units is stored at the intermediate nodes $\{I_z: z = 1, 2, \ldots, p\}$.

The PMSF problem with intermediate storage in a single source and multiple sinks network has been solved in [16]. A three layer algorithm has been presented to compute the priority ordering. The first layer gives the priority for sinks, second layer gives the priority for intermediate nodes and the third layer gives the combined priority ordering, i.e., (priority ordering of sinks, priority ordering of intermediate nodes). A polynomial time algorithm has been presented to solve the PMSF problem with intermediate storage in static network. For the single source and multiple sinks network, the priority order of the sinks is $\{D_y: y = 1, 2, \ldots, r\}$. At first, the minimum distance is determined for all $v \in I$ using algorithm in [5] and their priority ordering is fixed giving first priority to the node at farthest from the source. Then, for each $v \in I$ with $h(v) > 0$, we create virtual shelter $v' \in I'$ with the same priority ordering and consider them as sinks such that $u(v, v') = h(v) = h(v')$ and $c(v, v') = 0$. Creation of virtual sinks guarantee the flow conservation constraints at the intermediate nodes. We set the priority ordering of virtual sinks together with the given sinks as $(D_1, D_2, \ldots D_r, \ldots D')$ where $D \cup I' = D'$ and obtain the transformed single source and multiple sinks network as $\overline{N}^a = (V', E', u', h, c, s, D')$. In this network, the PMSF problem is solved using the algorithm of [14]. If the maximum value $(\max val\, D')$ is the amount of flow in to the sink set $D'$, then $(\max val\, D') = \sum_{x=1}^{q} f(S_x) = \sum_{y=1}^{r} f(D_y) + \sum_{z=1}^{p} f(I_z')$ where $\sum_{z=1}^{p} f(I_z')$ is the amount of flow in to the virtual sinks. Finally, the flow from the virtual sinks is transferred to the respective intermediate nodes by removing the virtual nodes and arcs. Removing the virtual parameters does not violate the feasibility conditions. Detail can be found in [16]. Following from this description we state Theorem 1.

**Theorem 1.** [16] There always exists an optimal solution to the prioritized maximum static flow problem with intermediate storage in a single source and multiple sinks network.

Now, we adopt the solution techniques presented in [17] for the maximum dynamic flow problem with intermediate storage allowing lane reversals in two terminal network in which the lexicographic property has been used to solve the problem and in [16] for the PMSF problem with intermediate storage in a single source and multiple sinks network. Then, we develop Algorithm 1 for multi-terminal network that solves the PMSF problem, Problem 1.

**Problem 1:** For a given multiple sources and multiple sinks network $N = (V, A, u, h, c, S, D, I)$, with the priority ordering on sources and sinks, the PMSF with intermediate storage allowing lane reversals is to obtain the feasible flow that maximizes the amount of flow out of the sources and the excess flow that does not reach the sinks can be stored at the intermediate nodes as far as possible from the sources if direction of the arcs can be reversed.

**Algorithm 1: The PMSF with intermediate storage allowing lane reversals**

**Input**: Given network $N = (V, A, u, h, c, S, D, I)$, with priority ordering of sources and sinks

**Output:** Prioritized maximum static flow with intermediate storage allowing lane reversals.

1. Obtain auxiliary static network $N^a = (V, E, u, h, c, S, D, I)$ with $u(e) = u(e') + u(e^{-1})$ $\forall e' \cup e^{-1} \in A, e \in E$.
2. For given prioritized sources $\{S_x: x = 1, 2, \ldots, q\}$, fix $S_1$ and solve the PMSF as in [16].
    i. Determine the minimum distance $dis(S_1, v)$ for each $v \in I$, using the algorithm in [5] and set their priority order giving the first priority to the node with the longest distance $dis(S_1, v)$.
    ii. Create virtual shelter $v' \in I'$ for each $v \in I$ and consider as sink with the same priority order as of $v \in I$.

iii. Obtain the transformed single source and multiple sinks network $\overline{N}^a = (V', E', u', h, c, S_1, D')$ considering $v' \in I'$ as sinks (priority ordering of given sinks is flowed by the virtual sinks).
iv. Compute priority based maximum flow without intermediate storage in $\overline{N}^a$ using procedure in [14].
v. Transform the solution of $\overline{N}^a$ in to $N^a$ by removing the virtual sinks and arcs using procedure in [17].

3. Fix $S_2$ and perform Steps (i) to (v) in Step 2 and continue for $\{S_x: x = 3, 4, \dots, q\}$.
4. Decompose the flow into paths and cycle and remove cycle with positive flow if any.
5. An arc $(w, v) \in A$ is reversed if and only if $f(v, w) > u(v, w)$ or if there is a nonnegative flow along the arc $(v, w) \notin A$.

**Theorem 2.** Algorithm 1 computes an optimal solution for PMSF problem with intermediate storage allowing arcs reversal in polynomial time complexity.

**Proof:** For the proof of this theorem, we have to show the feasibility and optimality of Algorithm 1. In Step 1, we redefine the arc capacity by summing the arc capacities in both directions, so it is feasible. The feasibility of Step 2 follows from Theorem 1. Steps 3 and 4 are feasible. As Step 4 does not contain any cycle, the flow in Step 5 is either in direction $(v, w)$ or in $(w, v)$ but never in both directions simultaneously. So, Step 5 is feasible.

For the optimality proof, first we construct the auxiliary network $N^a = (V, E, u, h, c, S, D, I)$ by adding two ways capacities of each arc. On the auxiliary network $N^a$, we fix the source with first priority as in Step 2 and solve Steps (i-v) as in [16,17] from which we get an optimal PMSF solution according to Theorem 1 on a single source and multiple sinks network. In the procedure, first we consider the sink $D_1$ and apply maximum flow algorithm for the maximum $s - d$ flow satisfying the conservation constraints. This $s - d$ flow is considered as initial flow and the residual capacities are updated for the gradual increment of flow and continue the procedure for other sinks. Finally, the flow accumulated in the virtual sinks is transferred in to the respective intermediate nodes using the procedure in [17]. Similarly, we fix another prioritized source and perform the same process and continue until there exists prioritized source. The PMSF solution obtained in the multiple sinks in auxiliary network is equivalent to the PMSF with lane reversals for original network $N$ according to [20] and the flow storage at intermediate nodes is same as in $N^a$, [17].

Now, the construction of auxiliary network $N^a$, setting priority ordering and creating virtual sinks $v' \in I'$ for all $v \in I$, require linear time. Similarly, the construction of a single source and multiple sinks virtual network $\overline{N}^a = (V', E', u', \square, c, s, D')$ requires linear time. Solution of the PMSF problem in the transformed network using the procedure in [14] can be computed in polynomial time complexity. Transformation of flow from the transformed network to the auxiliary network requires linear time. Irrelevant cycles can be removed by decomposing the flows in to paths and cycles. The decomposition requires polynomial time complexity and guarantee that flow is not in the both directions. All the steps in the algorithm have the polynomial time complexity. Repeated use of the solution procedure of [14] in Step 3 for $\{S_x, x = 2, \dots, q\}$ determines the complexity of the algorithm. In worst case, we may repeat for $n$ times. Hence, an optimal solution to the PMSF problem with intermediate storage allowing lane reversals can be obtained in polynomial time. ■

**Example 1:** Consider a multi-terminal two ways network containing two sources with priority order $s_2, s_1$ and two sinks with priority order $d_1, d_2$, as in the given Fig. 1(i). Here $x, y$ and $z$ are the intermediate nodes with capacities 22, 17 and 25 respectively. Each arc has capacity and cost, for example, arc $(s_1, y)$ has capacity 5 and traversing cost 1. After reversing the arcs and updating the arcs capacities, Fig. 1(ii) is obtained. Now, to compute the PMSF on this network, we fix source $s_2$ (as its order). Then, we calculate the shortest distance of each nodes $v \in I$ from $s_2$ and fix the priority ordering giving first priority to the node with longest distance from $s_1$ as $(z, y, x)$ and consider these nodes as virtual shelters as $x'$ $z'$ and $y'$ of $x$, $z$ and $y$, respectively so that we have $D' = D \cup \{x', z', y'\}$ as in Fig. 1(iii). Then in the next step, we push the flow to the sinks with the given priority ordering. The paths $s_2 - x - d_1$, $s_2 - x - y - d_1$, $s_2 - z - d_2$ carry 1, 2, 2 units flow, to sinks $d_1$ and $d_2$, respectively. As arcs $(x, d_1)$, $(x, y)$, $(s_2, z)$, are saturated, path $s_2 - x - x'$ carries 1 units flow at $x'$. Remove the virtual sink, transfer the flow in corresponding intermediate node and update the residual capacities of the network. Now, fix $s_1$ and repeat the same process. The paths $s_1 - y - d_1$, $s_1 - y - d_2$, and $s_1 - z - d_2$ carry 2, 3 and 1 units flow to the sinks $d_1$ and $d_2$, respectively. As arc $(s_1, y)$ is saturated, only path $s_1 - z - z'$ carries excess flow 6 units at $z'$. Again accumulate flow at intermediate nodes such that Fig. 1(iv) gives the optimal solution where 5 and 6 units of flows are reached at $d_1$ and $d_2$, respectively and 1,0 and 6 units of flows are stored in $x, y$ and $z$, respectively.

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 7 Issue 7, July 2020
**ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72**
www.ijiset.com

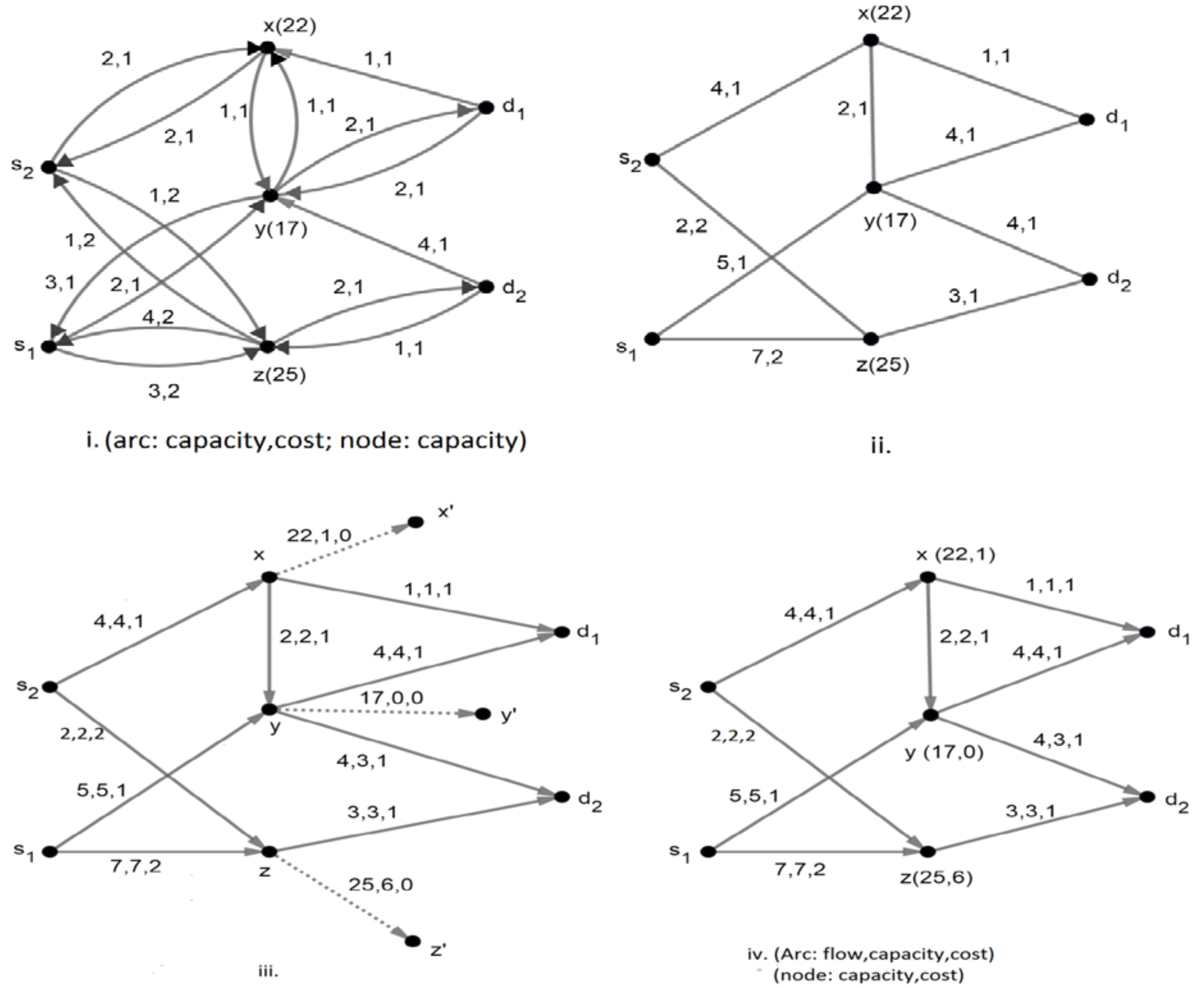


Fig. 1 i. Given network, ii. Auxiliary network, iii. PMSF on auxiliary network, iv. PMSF solution with intermediate storage.

## 4. Prioritized maximum dynamic flow with intermediate storage allowing lane reversals

In this section, we investigate the prioritized maximum dynamic flow (PMDF) problem with intermediate storage allowing lane reversals on multiple sources and multiple sinks network having given priority ordering of sources and sinks. This problem without intermediate storage has been introduced in [20] in which a polynomial time algorithm has been presented to solve it. On the other hand, the maximum dynamic flow and the earliest arrival flow problems with intermediate storage allowing lane reversals have been introduced and solved with polynomial time algorithms in [17,18]. To solve these problems, the former one in two terminal network, and later one in two terminal series-parallel network, the lexicographic property of [9,10] has been used. The two terminal networks have been converted into a single source and multiple sinks network with priority ordering and used the lexicographic property. We adopt these solution techniques to develop a polynomial time algorithm to solve the PMDF problem with intermediate storage with lane reversal capability independently.

**Problem 2:** For a given multiple sources and multiple sinks $N = (V, A, u, h, \tau, S, D, I, T)$ with given priority ordering on sources and sinks, the PMDF with intermediate storage allowing lane reversals is to obtain the feasible flow that maximizes the amount of flow out of the sources and the excess flow that does not reach the sinks within the given time horizon $T$ that can be stored at the intermediate nodes which are as far as possible from the sources.

Recently, the PMDF problem with intermediate storage without lane reversals has been solved in a single source and multiple sinks network, [16]. The priority ordering of sinks has been fixed giving the first priority for the sink at longest distance from the source, second priority for the sink with second longest distance and so on. Similarly, the priority ordering of intermediate nodes has been fixed and those nodes are treated as virtual sinks with respective priority so that the network is again converted into the transformed network having a single source and multiple sinks. Then, the PMDF [9,10] has been used to solve the problem on the transformed network in which total amount of flow in to the sinks is $\sum_{y=1}^{r} \lambda(D_y) + \sum_{z=1}^{p} \lambda(I'_z)$. The flow value $\sum_{z=1}^{p} \lambda(I'_z)$ in the virtual sinks should be transferred in to the intermediate nodes by removing virtual arcs and sinks as in [17]. Removing the virtual parameters does not violate the feasibility conditions. Let maximum value ($\max val\, D'$) be the amount of flow in to the sink set $D'$, then $(\max val\, D') = \sum_{x=1}^{q} \lambda(S_x) = \sum_{y=1}^{r} \lambda(D_y) + \sum_{z=1}^{p} \lambda(I_z)$ where $\sum_{z=1}^{p} \lambda(I_z)$ is the amount of flow in to the intermediate nodes. Detail can be found in [16, 17]. Thus, with these concepts Theorem 3 is stated.

**Theorem 3.** [16] The prioritized maximum dynamic flow problem with intermediate storage can be solved optimally in a single source and multiple sinks network.

Combining the PMDF flow algorithm with lane reversals of [20], maximum dynamic flow with intermediate storage allowing lane reversals of [17] and prioritized maximum dynamic flow of [16], we present Algorithm 2 that solves Problem 2.

**Algorithm 2: PMDF with intermediate storage allowing lane reversals**

**Input**: Given network $\boldsymbol{N = (V, A, u, h, \tau, S, D, I, T)}$ with the priority ordering of sources and sinks.

**Output:** PMDF with intermediate storage allowing lane reversals

1. Obtain auxiliary network $N^a = (V, E, u, h, \tau, S, D, I, T)$ with modified capacity and transit time as follows:

$$u(e) = u(e') + u(e^{-1}) \ \forall e' \cup e^{-1} \in A, e \in E$$
$$\tau(e) = \begin{cases} \tau(e') & \text{if } e' \in A \\ \tau(e^{-1}) & \text{otherwise} \end{cases}$$

2. For prioritized sources $\{S_x : x = 1, 2, \dots, q\}$, fix $S_1$ and solve the PMDF as in [16].
   i. Determine the minimum distance $dis(S_1, v)$ for each $v \in I$, using the algorithm in [5] and set their priority order giving the first priority to the node with longest $dis(S_1, v)$.
   ii. Create virtual shelter $v' \in I'$ for each $v \in I$ and consider as sink with the same priority order.
   iii. Obtain the transformed a single source and multiple sinks network $\overline{N}^a$ considering $v' \in I'$ as sinks in which the priority ordering of given sinks is followed by the virtual sinks.
   iv. Compute PMDF without intermediate storage in the transformed network $\overline{N}^a = (V', E', u', \square, \tau, S_1, D' T)$ according to [9, 10].
   v. Transform the solution of $\overline{N}^a$ in to $N^a$ by removing the virtual sinks and arcs.
3. Fix $S_2$ and perform Steps (i) to (v) in Step 2 and continue for $\{S_x : x = 3, 4, \dots, q\}$.
4. Decompose the flow in to paths and cycle and remove cycle with positive flow if any.
5. An arc $(w, v) \in A$ is reversed if and only if $f(v, w) > u(v, w)$ or if there is a nonnegative flow along arc $(v, w) \notin A$.

**Theorem 4.** Algorithm 2 computes an optimal solution for PMDF problem with intermediate storage allowing lane reversals in polynomial time complexity.

**Proof:** Proof of the theorem follows from the feasibility and optimality of Algorithm 2. In Step 1, we construct an auxiliary network in which arcs capacity are increased by combining the arc capacity in both directions so, it is feasible. Step 2 is feasible due to Theorem 3 and Step 3 is itself feasible. As decomposition of network flow into paths and cycle is feasible, Step 4 is feasible. Moreover, Step 4 guarantees that flow in Step 5 is either along $(v, w)$ or $(w, v)$ but not in the both directions showing the feasibility of Step 5.

For the optimality, at first we construct an auxiliary network $N^a = (V, E, u, h, \tau, S, D, I, T)$ where arc capacities are increased by using the arc capacity in both directions. We fix the source $S_1$ with first priority and solve Step 2 in $N^a$ as in [16,17] which gives PMDF and according to Theorem 3 such flow is optimal in a single source and multiple sinks network. The accumulated flow $\sum_{z=1}^{p} \lambda(I_z')$ in $\overline{N}^a$ is shifted to the corresponding intermediate nodes in $N^a$ using the procedure in [16,17] and proceed for other sinks. Similarly, we fix other sources according to their priority order and continue the process for $S = \{S_x : x = 2,3, \dots, q\}$. According to [20], the PMDF solution obtained in the auxiliary network $N^a$ is equivalent to the PMDF with lane reversals in the original network $N$ and according to [17], flow stored in the intermediate nodes is same as in $N^a$.
Computation of PMDF in the transformed network determines the complexity of the algorithm as construction of $N^a$, setting priority ordering and creating virtual sinks $v' \in I'$ for all $v \in I$, construction of $\overline{N}^a$ and transformation of accumulated flow from $\overline{N}^a$ to the corresponding intermediate node in $N^a$ require linear time. In Step 2(iv), prioritized maximum dynamic flow algorithm of [9,10] is applied. For this, a super terminal say $*$ is introduced and connected to each of the terminals $s_i$. If $s_i$ is a source then $u(*, s_i) = \infty$ and $\tau(*, s_i) = 0$ and compute minimum cost maximum flow from $*$ to $s_i$ in the residual network. Similarly, if $s_i$ is a sink then $u(s_i, *) = \infty$ and $\tau(s_i, *) = -(T+1)$ and compute minimum cost circulation in the residual network using $\tau$ as cost. The algorithm performs $k$ iterations for $i = k, k-1, \dots, 2,1$. The algorithm has time complexity $O(k(MCF))$ where $k$ is the number of terminals and $O(MCF)$ is the complexity to perform minimum cost circulation. To avoid the irrelevant cycles, flow is decomposed in to paths and cycles that can be done in polynomial time complexity and guarantee that flow is along single direction. Repeated use of the algorithm in [9,10] in Step 3 for $x = 2, 3, \dots, q$ does not effect the worst case time complexity. Hence, Algorithm 2, computes the PMDF with intermediate storage allowing lane reversals in polynomial time complexity. ■

**Example 2:** If we consider the cost as a time in Fig. 1 (i), the static network can be changed in to dynamic network. Then after reversing the arcs and updating the arcs capacities and traversing times, the Fig. 1(ii) is obtained. As in static network, we start from the source $s_2$ and after creating the virtual shelters, send the prioritized maximum dynamic flows on that network by giving first priority to sinks, after that the shelter which is at the longest distance from $s_2$ and so on. Repeat the same process to the source $s_1$. Here we have $T = 3$. The dynamic solution for Fig. 1(iii) is given by Table 1.

**Table 1.** PMDF with intermediate storage allowing lane- reversals

| S.N. | Path | Time - 1 | Time - 2 | Time - 3 | Total |
|---|---|---|---|---|---|
| 1 | $s_2 - x - d_1$ | -- | 1 | 1 | 2 |
| 2 | $s_2 - x - y - d_1$ | -- | -- | 2 | 2 |
| 3 | $s_2 - z - d_2$ | -- | -- | 2 | 2 |
| 4 | $s_2 - x - y - y'$ | -- | -- | 2 | 2 |
| 5 | $s_2 - z - z'$ | -- | -- | 2 | 2 |
| 6 | $s_2 - x - x'$ | 1 | 1 | 4 | 6 |
| 7 | $s_1 - y - d_1$ | -- | 2 | 2 | 4 |
| 8 | $s_1 - y - d_2$ | -- | 3 | 3 | 6 |
| 9 | $s_1 - z - d_2$ | -- | -- | 1 | 1 |
| 10 | $s_1 - z - z'$ | -- | 6 | 7 | 13 |
| 11 | $s_1 - y - y'$ | -- | -- | 5 | 5 |

From the Table 1, we see that in time period $T = 3$, sinks $d_1$ and $d_2$ receive 8 and 9 units of flow. Similarly, 6, 7 and 15 units flow are stored at the intermediate nodes $x, y$ and $z$.

## 5. Conclusions

From literature, the flow models allowing lane reversals without and with intermediate storage have been studied. In this paper, we have considered the multiple sources and multiple sinks networks with given priority ordering and investigated the flow models with intermediate storage allowing lane reversals. The prioritized maximum static flow problem with intermediate storage has been introduced and presented a polynomial time algorithm to solve the problem. Similarly, the prioritized maximum dynamic flow problem with intermediate storage allowing lane reversals has been introduced and solved it using polynomial time algorithm. When flow is pushed saturating all arcs out of the sources, total amount of flow can be more than the capacity of the minimum cut. If the minimum cut capacity of the network is symmetric, then lane reversal capability increases the prioritized maximum flow in static and dynamic networks upto doubled at each prioritized sinks. Moreover, the excess flow value also increases at the intermediate nodes having sufficient capacities to store flow.

To the best of our knowledge, we have introduced and solved these problems for the first time. Further, we are interested in the flow management stored at the intermediate node.

**Acknowledgments**

The first author thanks Alexander von Humboldt Foundation for her return fellowship (November 2019-October 2020) at Central Department of Mathematics, TU, Nepal. The second author thanks UGC Nepal for the MPhil fellowship.

## References

[1] Andreas A.K. and Smith J.C. Decomposition algorithms for the design of a non simultaneous capacitated evacuation tree network. Networks: An International Journal, 2009, 53(2), pp. 91-103.

[2] Arulselvan A. Network model for disaster management. PhD Thesis, University of Florida, USA, 2009.

[3] Achrekar O. and Vogiatzis C. Evacuation trees with contraflow and divergence considerations. In International Conference on Dynamics of Disasters, 2018, pp.1-46. Springer, Cham.

[4] Dhamala T.N., Pyakurel U. and Dempe S. A critical survey on the network optimization algorithms for evacuation planning problems. International Journal of Operations Research, 2018, 15(3), pp.101-133.

[5] Dijkstra E.W. A note on two problems in connection with graphs. Numerische Mathematik, 1959,1(3), pp. 269-271.

[6] Ford F.R. and Fulkerson, D.R. Maximal flows through a network. Canadian Journal of Mathematics,1956, 8, 399 – 404.

[7] Ford F.R. and Fulkerson, D.R. Flows in Networks. Princeton University Press, Princeton, New Jersey, USA, 1962.

[8] Harris T.E. and Ross F.S. Fundamentals of method for evaluating rail net capacities . Research Memorandum, 1955.

[9] Hoppe B. and Tardos E. Polynomial time algorithms for some evacuation problems. Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 433-441.

[10] Hoppe B. and Tardos E. The quickest transshipment problem. Mathematics of Operations Research, 2000, 25:36-62.

[11] Khadka S.R. and Bhandari P.P. Model and solution for non-conservation flow evacuation planning problem. The Nepal Math. Sc. Report, 2019, 36(1& 2), pp. 11-16.

[12] Kim S., Shekhar S. and Min M. Contraflow transportation network reconfiguration for evacuation route planning. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(8):1115-1129.

[13] Kotsireas I.S., Nagurney A. and Pardalos P. M. (Eds) Dynamics of disasters-algorithmic approaches and applications. Springer Optimization and Its Applications 2018.

[14] Minieka E. Maximal, lexicographic, and dynamic network flows. Operations Research, 1973, 21:517-527.

[15] Nath H.N., Pyakurel U., Dhamala T.N. and Dempe S. Dynamic network flow location models and algorithms for evacuation planning. Journal of Industrial and Management Optimization, 2020, doi:10.3934/jimo.2020102.

[16] Pyakurel U. and Adhikari M.C. Priority Based Flow Improvement with Intermediate Storage. Submitted to International Journal of Operations Research, 2020, (cf. Optimization Online http://www.optimization-online.org/DB_HTML/2020/07/7891.html).

[17] Pyakurel U. and Dempe S. Network flow with intermediate storage: Models and algorithms. Preprint, 01/2009 TU Bergakademie Freiberg 2019.

[18] Pyakurel U. and Dempe S. Universal maximum flow with intermediate storage for evacuation planning. 4th International conference on Dynamics of Disasters, July 1-4, Kalamata, Greece, 2019 (to appear in Dynamics of Disasters 2020).

[19] Pyakurel U. and Dhamala T.N. Contraflow emergency evacuation by earliest arrival flow. PhD progress report presentation at University Grant Commission Nepal 2012. www.researchgate.net/profile/Urmila_Pyakurel3/publications.

[20] Pyakurel U. and Dhamala T.N. Models and algorithms on contraflow evacuation planning network problems. International Journal of Operations Research, 2015, 12(2), pp. 36-46.

[21] Pyakurel U., Dhamala T.N. Continuous time dynamic contraflow models and algorithms. Advances in Operations Research - Hindawi; Article ID 368587, 2016, pp. 1-7.

[22] Pyakurel U., Dhamala T.N. and Dempe S. Efficient continuous contraflow algorithms for evacuation planning problems. Annals of Operations Research (ANOR), 2017, 254, pp. 335-364.

[23] Pyakurel U., Dhamala T.N. Continuous dynamic contraflow approach for evacuation planning. Annals of Operation Research, 2017, 253, pp. 573-598.

[24] Pyakurel U., Nath H.N. and Dhamala T.N. Efficient contraflow algorithms for quickest evacuation planning, Science China Mathematics, 2018, 61, pp. 2079-2100.

[25] Pyakurel U., Nath H.N., Dhamala T.N. Partial contraflow with path reversals for evacuation planning. Annals of Operations Research, 2019, 283, pp. 591-612

[26] Pyakurel U., Nath H.N., Dempe S. and Dhamala T.N. Efficient dynamic flow algorithms for evacuation planning problems with partial lane reversal, 2019, Mathematics, 7, pp. 1-29.

[27] Rebennack S., Arulselvan A., Elefteriadou L. Pardalos P. M. Complexity analysis for maximum flow problems with arc reversals. Journal of Combinatorial Optimization, 2010, 19, pp. 200-216.

[28] Ruzika S., Sperber H., Steiner M. Earliest arrival flows on series- parallel graphs. Networks, 2011, 10, pp. 169-173.

[29] Vogiatzis C., Walteros J. L. and Pardalos P.M. Evacuation through clustering techniques. In: Goldengorin B., Kalyagin V., Pardalos P. (eds) Models, Algorithms, and Technologies for Network Analysis. Springer Proceedings in Mathematics and Statistics, vol 32. Springer, New York, NY, 2013.

[30] Zheng Q.P., Arulselvan A. Discrete time dynamic traffic assignment models and solution algorithm for managed lanes. Journal of Global Optimization, 2011, 51(1), pp. 47-68.

**First Author** Urmila Pyakurel completed her M.Sc in 2007 and PhD in Optimization under the supervision of Prof. Dr. Tanka Nath Dhamala in 2016 from Tribhuvan University. From Nepal Academy of Science and Technology, she received the Young Scientist Award in 2016. Dr. Pyakurel has visited the University of Kaiserslautern, Germany in 2014 for perusing her PhD research under the invitation of Prof. Dr. Horst W. Hamacher. Under the invitation of Prof. Dr. Stephan Dempe, she visited TU Bergakademie Freiberg, Germany as a researcher under the Alexander von Humboldt Foundation Research Group Linkage Program, in the year 2016. She worked as a Post-doctoral Research Fellow with Prof. Dr. Stephan Dempe under Georg Foester Fellowship of the Alexander von Humboldt Foundation at TU Bergakademie Freiberg during November 2017 - October 2019. Currently she is working as an Associate Professor, in Central Department of Mathematics, Tribhuvan University, Kathamndu, Nepal and is involved in teaching and research activities including the DAAD and AvH research programs in Optimization and Graph Theory in the department. Dr. Pyakurel has published a number of papers in the journals of repute and participated in national and international conferences and workshops. She has been academically affiliated in many academic organizations in Nepal and abroad like AMS, MDMC, NEGAAS and NMS. She has served as treasurer of Nepal Mathematical Society and is also a life-member of the society. Mathematical modeling and solving the real-life problems using optimization, simulation and heuristic algorithms, specially the contarflow algorithms in evacuation planning, logistics and location analysis are the main research areas of Dr. Pyakurel.

**Second Author** Sachin Wagle received his M.Sc. Degree from Central Department of Mathematics, Tribhuvan University in 2003. Now he is MPhil scholar at Central Department of Mathematics. Currently he is working at IOE Purwanchal Campus, Dharan, Sunsari, Nepal.

**Third Author** Mohan Chandra Adhikari received his MA Degree from Central Department of Mathematics, Tribhuvan University in 2009. Now he is MPhil scholar at Central Department of Mathematics. Currently he is working at National School of Sciences, Lainchour, Kathmandu, Nepal.