The Mathematical Sciences Report, Vol. 39, No. 2, 2022: 67-78
DOI: https://doi.org/10.3126/nmsr.v39i2.51695

ABSTRACT TEMPORALLY REPEATED FLOW WITH INTERMEDIATE STORAGE

DURGA PRASAD KHANAL[1], URMILA PYAKUREL[2], TANKA NATH DHAMALA[3], STEPHAN DEMPE[4]

[1] *Saraswati Multiple Campus, Tribhuvan University, Kathmandu, Nepal;*
[2,3] *Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal;*
[4] *Faculty of Mathematics and Computer Science, TU Bergakademie Freiberg, Freiberg, Germany;*
[2] *Corresponding author: urmilapyakurel@gmail.com*

Abstract: Network associated with the set of elements and linearly ordered subset of elements, known as paths, satisfying the switching property is an abstract network. Due to the switching property, flows crossing at intersections are diverted to the non-crossing sides. Each element of an abstract network is equipped with two types of integral capacities: one is movement capacity which transships the flow from an element to its adjacent element and another is the storage capacity which holds the flow at the element. Due to insufficient movement capacity of intermediate elements, flow out from the source may not reach at the destination. If the flow out from the source is more than the minimum cut capacity, then the problem associated with the settlement of excess flow at appropriate intermediate elements is termed as network flow with intermediate storage. In this paper, we discuss the static and dynamic flow models with intermediate storage in an abstract network using temporal repetition of flow. We solve abstract maximum dynamic flow and contraflow problems with intermediate storage.

Keywords: Abstract network, maximum flow, switching property, temporally repeated flow, contraflow

1. Introduction

A disaster is a disruption occurring over a short or long period of time that causes massive loss of human, material, economic or environment. A very efficient post disaster evacuation planning is essential to minimize the losses. At the time of evacuation, evacuees are to be shifted from danger zones (sources) to safety places (sinks) as quickly and efficiently as possible. In contrast to evacuation planning problems based on the flow conservation constraints at intermediate elements, evacuation planning problem with intermediate storage deals with the settlement of excess flow at intermediate elements. By holding the excess flow not reaching to the destination at comparatively safer intermediate shelters, it maximizes the number of evacuees leaving the danger zone. To address such problems in classical network, Pyakurel and Dempe [19] introduced the concept of network flow with intermediate storage and presented polynomial time algorithm to solve the maximum dynamic flow problem. Pyakurel and Dempe [21] investigated universal maximum dynamic flow with intermediate storage and presented efficient algorithms in general as well as two-terminal

Received: November, 2022 Accepted/ Published Online: December, 2022.

series parallel networks. To transship more than one commodity, Khanal et al. [11] introduced the maximum multicommodity flow problem with intermediate storage and presented efficient algorithm to solve it. Not only in evacuation planning, intermediate storage is highly applicable for different demand-supply chains like commodity supply, electricity distribution, water supply, etc.

In two way network topology, an important scenario after disaster is that the paths towards the risk zones are almost empty (i.e., no flow on paths towards the danger zones). The optimal use of empty paths to maximize the flow and minimize the time of evacuation is possible by a widely accepted technique, known as contraflow configuration. Kim et al. [15] presented a greedy heuristic to produce high quality solutions and a bottleneck heuristic to deal with large scale evacuations. The strongly polynomial time algorithms for maximum and quickest contraflow problems in two terminal network with discrete time settings can be found in Rebenack et al. [27]. In continuous-time settings, Pyakurel and Dhamala [20] introduced the dynamic contraflow model. By using the natural transformation of Fleischer and Tardos [3], they have presented efficient algorithms to solve the maximum, quickest, and earliest arrival flow problems with lane reversals. Pyakurel et al. [25] introduced the concept of partial lane reversals in which only necessary arc capacities are reversed to increase the flow value and unused arc capacities are saved for other emergency purposes like logistic supports and facility locations. Models and solution strategies of different problems regarding the evacuation plannings with/without contraflow can be found in [1, 6, 10, 12, 14, 23].

A network with capacitated elements and linearly ordered subset of elements, called paths, is an abstract network. When two paths in abstract network cross at an element then there must be a path that is a subset of the first path up to the crossing element and a subset of second path after the crossing element, known as switching property. Hoffman [7] introduced the concept of abstract flow by reviewing the first proof of max-flow-min-cut theorem of Ford and Fulkerson [4] with flows in term of paths rather than on arcs. McCormick[18] provided a polynomial time algorithm by using an oracle where input is an arbitrary subset of elements whose output is either a path contained in that subset or states that no such path exists. He used the augmenting path structure satisfying the complementary slackness condition: every positive path meets the cut set exactly at one common element and every element of the cut is saturated. Martens and McCormick [17] extended the result of [18] in more general case by using additional attribute of weight on paths. Martens [16] presented unsplittable and k-splitable abstract network flows. Similarly, Kappmeier [9] presented polynomial algorithm for lexicographic abstract maximum flow and used it to prove the existence of abstract earliest arrival flow.

The concept of continuous maximum abstract contraflow problem is introduced by Pyakurel et al. [22]. They have presented polynomial time algorithms to solve the problems. Similarly, Pyakurel et al. [26] introduced the partial contraflow approach in abstract network by saving unused capacities of the elements, and presented efficient algorithms for static, lexicographically maximum static, maximum dynamic and earliest arrival partial contraflow problems. The polynomial time algorithms to solve the maximum static, lexicographic maximum static and maximum dynamic flow problems in abstract network and abstract contraflow network with intermediate storage can be found in Pyakurel et al. [24]. The have introduced the formula of temporally repeated flow to solve maximum dynamic flow with intermediate storage if the holding capacity of nodes is sufficient.

By introducing the partial switching property, Khanal et al. [13] solved an abstract quickest flow problem with partial switching of paths.

For a given set of source-sink paths with fixed transit times, if the flow is sent along the decomposed paths repeatedly over the time with constant rate of flow, is called temporally repeated flow. Ford and Fulkerson [5] introduced the temporally repeated flow in which a stationary maximal dynamic flow can be obtained by solving a transshipment problem associated with the static network. This flow is known to be an optimal dynamic flow. Using the concept of temporally repeated flow in classical network, Kappmeier et al. [8] presented the temporally repeated abstract flow to obtain abstract flow over time from source to the sink. Here, we introduce the temporally repeated flow on an abstract network from the source element to the sink as well as each of the intermediate elements. With the help of intermediate storage, it maximizes the flow out from the source in polynomial time complexity.

In this paper, we present mathematical models for static and dynamic flow problems with intermediate storage. We introduce a temporally repeated flow with intermediate storage to obtain the abstract maximum dynamic flow (MDF) if the storage capacity of each intermediate element is sufficient (i.e., T times the sum of incoming capacities from its left elements through the paths) and present a polynomial time algorithm to solve MDF problem. We also present a maximum dynamic contraflow (MDCF) problem with intermediate storage using the temporally repeated flow and present polynomial time solution strategy. We organize the paper as follows. Section 2 provides the basic definitions and mathematical formulations of the flow models. In Section 3, we introduce abstract temporally repeated flow to solve the maximum dynamic flow problem with intermediate storage. Similarly, we solve abstract temporally repeated MDCF problem with intermediate storage in Section 4. The paper is concluded in Section 5. To the best of our knowledge, problems introduced in Section 3 and Section 4 and their solution strategies with temporally repeated flow are investigated for the first time.

2. Mathematical Formulation of Flow Models

In this section, we give basic mathematical denotations that are used throughout the paper. We also present static as well as dynamic flow models for abstract network with intermediate storage.

2.1. **Basic Denotations.** Consider a network $\mathcal{N} = (E, \mathcal{P})$ with finite set of elements E and the collection of paths

$$\mathcal{P} = \{P \subseteq E : P \text{ has a linear order } <_P \text{ of elements in } P\} \subseteq 2^E.$$

The collection $\mathcal{P}$ of paths contains source-sink (s-t) paths P as well as intermediate paths $P_{[s\to e]}$ from source $s \in E$ to an intermediate element $e \in E$. Each element $e \in E$ has the non-negative integral movement capacity $u_e : E \to \mathbb{Z}^+$ which is used to send flow from the element e to its adjacent element and the storage capacity $v_e : E \to \mathbb{Z}^+$ which is used to hold flow at e. The order of elements in the path $P \in \mathcal{P}$ is denoted by $<_P$. An element $a \in P$ is left of e on P if $a <_P e$ and right of e if $a >_P e$. Similarly, $e \in P$ is said to be leftmost or first (rightmost or last) element of P if there does not exist a in P such that $a <_P e$ $(a >_P e)$. For s-t path P, source s is the leftmost element and sink t is the rightmost element. Let $E_I = E \setminus \{s,t\}$ be the set of intermediate elements.

Network $\mathcal{N} = (E, \mathcal{P})$ is an abstract network if it satisfies the switching property: $\forall P, Q \in \mathcal{P}$ and intermediate element $e \in P \cap Q$, $\exists R \in \mathcal{P}$ such that $R \subseteq P \times_e Q$
where,

$$P \times_e Q = \{a \in P : s \leq_P a \leq_P e\} \cup \{a \in Q : e \leq_Q a \leq_Q t\}.$$

Similar definition for switched path $R \subseteq Q \times_e P$ can be obtained. For simplicity, we use the notations

$$P_{[s \to e]} = \{a \in P : s \leq_P a \leq_P e\} \text{ and } P_{[e \to t]} = \{a \in P : e \leq_P a \leq_P t\}$$

to represent the elements on path P from source s up to e and that begins from e up to sink t, respectively. Similarly,

$$P_{[s \to e)} = \{a \in P : s \leq_P a <_P e\}, \text{ and } P_{(e \to t]} = \{a \in P : e <_P a \leq_P t\}$$

represent the elements on path P that are left of e and right of e, respectively. If P and Q are two paths both containing e_1 and e_2, then it is possible to have $e_1 <_P e_2$ but $e_1 >_Q e_2$.

Throughout the paper, we consider that the storage capacity at source and sink elements are sufficiently large, i.e., $v_s = v_t \leq \infty$ and that of intermediate elements are finite. The movement capacity of source and intermediate elements are finite (i.e., $u_s < \infty$) and that of sink is zero (i.e., $u_t = 0$). If the incoming movement capacity of an intermediate element $e \in E_I$ is more than the outgoing movement capacity, then the excess flow is used to store at e. Moreover, for the uniqueness of the solution, the storage capacity of $e \in E_I$ should be $v_e \geq \sum_{P \in \mathcal{P}: a \in P} u_a \;\; \forall a <_P e$. Furthermore, the incoming and outgoing movement capacities of source and sink are zero, respectively, except for contraflow network.

2.2. **Abstract Static Flow Model with Intermediate Storage.**

Consider an abstract network $\mathcal{N} = (E, \mathcal{P})$ with path-flow $x_P : P \to \mathbb{R}^+$. Every path-flow x_P induces a flow through each element, denoted by $x_{P(e)} = \sum_{P \in \mathcal{P}: e \in P} x_P$. For all $e \in E$, if $x_{P(e)} \leq u_e$ and $x_P \geq 0$ then the path-flow x_P is feasible and an element e is said to be saturated with respect to x if $x_{P(e)} = u_e$. We denote $x_e^{out} = \sum_{P \in A_e} x_P$ and $x_e^{in} = \sum_{P \in B_e} x_P$ as the total outflow from e and the total inflow into e, respectively, where A_e and B_e represent the set of outgoing paths from e and incoming paths into e. Let $c_e : E \to \mathbb{Z}^+$ be the cost of transmission of flow per unit from e to its right element so that $c_P = \sum_{e \in P} c_e$.

Hoffman [7] generalized the max-flow-min-cut theorem of Ford and Fulkerson [4] for abstract network flow without storage of flow at intermediate elements. Due to the bottleneck flow on each paths, sending the flow with full movement capacity from source element greater than the minimum cut capacity is impossible. To deal with this problem, Pyakurel and Dempe [19] introduced the concept of intermediate storage for classical network flow. Using this concept in abstract network flow, we aim to push the maximum flow outward from the source in which flow with minimum cut capacity reaches to the sink and rest of the flow is to be stored at intermediate elements.

We define the flow function $x_e : E_I \to \mathbb{R}^+$ as the excess flow stored at element $e \in E_I$. The linear program for abstract static network flow with intermediate storage, as presented in [24], is

$$\max \quad \sum_{P\in\mathcal{P}} x_P + \sum_{e\in E_I} x_e \tag{2.1}$$

$$\text{s.t.} \quad \sum_{P\in\mathcal{P}:e\in P} x_P \le u_e \quad \forall e \in E \tag{2.2}$$

$$0 \le x_e^{in} - x_e^{out} = x_e \le v_e \ \ \forall e \in E_I \tag{2.3}$$

$$x_P \ge 0 \quad \forall P \in \mathcal{P} \tag{2.4}$$

$$v_e \ge \sum_{P\in\mathcal{P}:a\in P} u_a \ \ \forall e \in E_I, \ \ a <_P e \tag{2.5}$$

Equation (2.1) is an objective function that refers to maximize the total flow reaching at sink t and the excess flow stored at intermediate elements. Equation (2.2) represents the capacity constraint of each element. The left inequality of Equation (2.3) represents the non-conservation of flow whose right inequality indicates that the excess flow is bounded by the storage capacity of e. Similarly, Equation (2.4) represents the non-negativity of the flow on each path and lower bound of storage capacity is given in Equation (2.5).

2.3. **Abstract Dynamic Flow Model with Intermediate Storage.** Let us consider $\mathcal{N} = (E, \mathcal{P}, \tau, T)$ as the abstract dynamic network, where $\tau : E \to \mathbb{Z}^+$ be a non-negative transit time of element $e \in E$ that is necessary to transship the flow from e to its right element and $T \in \mathcal{T}$ be a time horizon. If e and a are two consecutive elements on path P with $e <_P a$, then flow traveling through e at time θ reaches a at time $\theta + \tau_e$. In discrete time setting, time horizon is discretized as $\mathcal{T} = \{0, 1, \ldots, T\}$. For each s-t path $P \in \mathcal{P}$, $\tau_P = \sum_{a\in P_{[s\to t)}} \tau_a$ denotes the traversal time of flow from s to t and $\tau_{P_{[s\to e]}} = \sum_{a\in P_{[s\to e)}} \tau_a$ denotes the traversal time of flow from s to intermediate element e through the path $P_{[s\to e]}$.

Let $\psi_P(\theta) : P \times T \to \mathbb{R}^+$ be the dynamic s-t path-flow in discrete time $\theta \in \mathcal{T}$ and $\psi_e(\theta) : E_I \times T \to \mathbb{R}^+$ be the amount of excess flow stored at intermediate element $e \in E_I$ within time $\theta \in \mathcal{T}$. Let $\psi_e^{out} = \sum_{P\in A_e} \psi_P$ and $\psi_e^{in} = \sum_{P\in B_e} \psi_P$ denote the total outflow from e and inflow into e, respectively. As in [24], the abstract dynamic flow model with intermediate storage is

$$\max \quad \sum_{\theta=\tau_P}^{T} \sum_{P\in\mathcal{P}} \psi_P(\theta) + \sum_{\theta=\tau_{P_{[s\to e]}}}^{T} \sum_{e\in E_I} \psi_e(\theta) \tag{2.6}$$

$$\text{s.t.} \quad \sum_{P\in\mathcal{P}:e\in P} \psi_P(\theta) \le u_e(\theta) \quad \forall e \in E, \ \theta \in \mathcal{T} \tag{2.7}$$

$$0 \le \psi_e^{in}(\theta) - \psi_e^{out}(\theta) = \psi_e(\theta) \le v_e \ \ \forall e \in E_I, \ \ \theta \in \mathcal{T} \tag{2.8}$$

$$\psi_P \ge 0 \quad \forall P \in \mathcal{P} \tag{2.9}$$

$$v_e \ge T \sum_{P\in\mathcal{P}:a\in P} u_a \ \ \forall e \in E_I, \ \ a <_P e, \tag{2.10}$$

The objective function in Equation (2.6) is to maximize the total flow reaching at t and the excess flow stored at intermediate elements within time horizon T. Similarly, Equation (2.7) represents the capacity constraint of each element at $\theta \in \mathcal{T}$ and Equation (2.9) represents the non-negativity of the flow on each path. Non-conservation of the flow at each time step θ is presented by the left inequality of Equation 2.8 whose right inequality shows that the excess flow is bounded by the storage capacity of e at each time $\theta \in \mathcal{T}$. The lower bounds of storage capacity of intermediate elements is represented in Equation (2.10) which is essential for the existence of temporally repeated flow.

3. Abstract Temporally Repeated MDF with Intermediate Storage

For a dynamic network $\mathcal{N}$, let u and τ be non-negative capacity and transit time, respectively, that are assigned to transship the flow from one element to its adjacent element via some path P. As in Kappmeier [9], we define the the time expanded ground set $\mathcal{E}_T$ by creating $T+1$ copies of elements for each time step $\theta \in \mathcal{T} = \{0, 1, \ldots, T\}$ as

$$\mathcal{E}_T = \{e^\theta : e \in E, \theta \in \mathcal{T}\}.$$

Flow starting from the source element at time θ reaches to an intermediate element e along path P at $\theta + \sum_{a \in P_{[s \to e)}} \tau_a$. The set of temporal paths P^θ associated with path $P \in \mathcal{P}$ that start from source element at time step θ is

$$P^\theta_{[s \to e]} = \left\{ e^\beta \in \mathcal{E}_T : e \in E, \beta = \theta + \sum_{a \in P_{[s \to e)}} \tau_a \right\}.$$

For the convenient notation, we use P^θ instead of source-sink temporal paths $P^\theta_{[s \to t]}$. The order of elements in temporal path is same as in original path P. We represent the set of all paths reaching to the intermediate element e and the sink element t within time horizon T by $\mathcal{P}^\theta_{[s \to e],T}$ and $\mathcal{P}^\theta_T$, respectively, and defined as

$$\mathcal{P}^\theta_{[s \to e],T} = \left\{ P^\theta_{[s \to e]} : P_{[s \to e]} \subset P \in \mathcal{P},\ \theta \in \mathcal{T},\ \theta + \sum_{a \in P_{[s \to e)}} \tau_a \leq T \right\}$$

and

$$\mathcal{P}^\theta_T = \left\{ P^\theta : P \in \mathcal{P},\ \theta \in \mathcal{T},\ \theta + \sum_{e \in P} \tau_e \leq T \right\},$$

Since the paths in $(\mathcal{E}_T, \mathcal{P}^\theta_T)$ may not satisfy the switching property, Kappmeier [9] presented time expanded network to obtain abstract maximum dynamic flow where paths P^δ with delay pattern δ are used to satisfy the switching property. Using this concept, we define the set of temporal paths with delay pattern δ arriving at the intermediate element e and destination sink t

within time T as

$$\mathcal{P}^{\delta}_{[s\to e],T} = \left\{ P^{\delta}_{[s\to e]} : P_{[s\to e]} \subset P \in \mathcal{P},\ \delta \in \{0,1,\dots,T\}^{P},\ \sum_{a\in P_{[s\to e)}} (\tau_a + \delta_a) \leq T \right\}$$

and

$$\mathcal{P}^{\delta}_{T} = \left\{ P^{\delta} : P \in \mathcal{P},\ \delta \in \{0,1,\dots,T\}^{P},\ \sum_{e\in P} (\tau_e + \delta_e) \leq T \right\},$$

respectively.

Lemma 3.1 ([9]). *If an abstract network $\mathcal{N} = (E, \mathcal{P})$ preserves the order on paths, then the network $\mathcal{N}_T = (\mathcal{E}_T, \mathcal{P}^{\delta}_{T})$ with path system $\mathcal{P}^{\delta}_{T}$ is an abstract network.*

Now, we introduce the abstract maximum dynamic flow problem with intermediate storage as follows.

Problem 1. *For a given abstract dynamic network $\mathcal{N} = (E, \mathcal{P}, \tau, T)$, an abstract maximum dynamic flow problem with intermediate storage is to find the maximum flow leaving the source element that is to be sent to the sink via s-t paths $P \in \mathcal{P}$ by allowing the maximum storage of excess flow at intermediate elements e via paths $P_{[s\to e]}$ $\forall e \in E_I$ with storage capacity $v_e \geq T \sum_{P\in\mathcal{P}:a\in P} u_a \ \ \forall a <_P e$ within the given time horizon T.*

To solve the problem, we begin the procedure by fixing the priority of elements. As in Pyakurel and Dempe [19], first priority is given to the sink to transship as much flow as possible. The excess flow is to be stored at the intermediate elements with priority order as follows: For each $e \in E_I$ with storage capacity $v_e \geq T \sum_{P\in\mathcal{P}:a\in P} u_a \ \ \forall a <_P e$, calculate the shortest distance $d_{P_{[s\to e]}}$ from s by using algorithm of Dijkstra [2]. It is to be noted that the temporally repeated solution exists only if the elements have upper bound capacities. The minimum cost path is considered as the shortest path and the priority is given to the farthest element among the elements of shortest distance. For example, if $d_{P_{[s\to e_1]}} > d_{P_{[s\to e_2]}}$ for $e_1, e_2 \in E_I$, then e_1 is higher in priority than e_2 and is denoted by $e_1 \succ e_2$.

Let $e_1, \dots, e_r$ be r intermediate elements with priority order $e_1 \succ e_2 \succ \cdots \succ e_r$ then set of all prioritized element including sink is $D = \{t \succ e_1 \succ \cdots \succ e_r\}$. For the notational convenient, if we write $t = e_0$ then set of prioritized elements becomes $D = \{e_0 \succ e_1 \succ \cdots \succ e_r\}$. As a solution strategy, we introduce the temporally repeated solution with intermediate storage and present an algorithm to solve Problem 1 herein.

3.1. **Temporally Repeated Flow with Intermediate Storage.** Temporally repeated flow, introduced by Ford and Fulkerson, induces the maximum dynamic flow from the source to the sink by repeating the constant rate of flow on source-sink paths with fixed transit times. Due to the flow conservation constraint, every flow out from the source may not reach at the sink element. Thus, the settlement of excess flow at intermediate elements plays an important role in evacuation planning, [19]. To deal with this problem, we introduce the temporally repeated flow in abstract network that transship the flow to the sink and intermediate elements with priority order which maximizes the flow out from the source element.

As the storage capacity of each prioritized intermediate element is $v_e \geq T \sum_{P \in \mathcal{P}: a \in P} u_a \quad \forall e \in E_I, \;\; a <_P e$, the temporally repeated flow on the sink and intermediate elements can be obtained through the paths with waiting pattern δ as follows, [24]:
For $e_0 = t$ and $P \in \mathcal{P}_T^\delta$ with $\psi_P = \min\{u_a : a \in P\}$,

$$|\psi|_{e_0,T} = \sum_P (T - \tau_P + 1).\psi_P \tag{3.1}$$

For intermediate element $e_i \in E_I$ and $P_{[s \to e_i]} \in \mathcal{P}^\delta_{[s \to e_i],T} \subseteq \mathcal{P}_T^\delta$ with $e_i <_P e_j$,

$$|\psi|_{e_i,T} = \sum_{P_{[s\to e_i]}} \Big[(T - \tau_{P_{[s\to e_j]}} + 1).\psi_{e_i} + (\tau_{P_{[s\to e_j]}} - \tau_{P_{[s\to e_i]}}).\psi_{P_{[s\to e_i]}}\Big] \tag{3.2}$$

where, $|\psi|_{e_i,T}$ is the net flow at element e_i within time T, $\psi_{P_{[s\to e_i]}} = \min\{u_a : a \in P_{[s\to e_i)}\}$ and $\tau_{P_{[s\to e_i]}} = \sum_{a \in P_{[s\to e_i)}} (\tau_a + \delta_a)$.

Now we present an algorithm to solve Problem 1.

Algorithm 1: Abstract temporally repeated MDF algorithm with intermediate storage

Input : Given abstract dynamic network $\mathcal{N} = (E, \mathcal{P}, \tau, T)$.

Output: Abstract temporally repeated MDF with intermediate storage on $\mathcal{N}$.

(1) Compute the shortest distance $d_{P_{[s\to e]}}$ $\forall e \in E$ by taking transit time as cost and using Dijkstra's algorithm.
(2) Fix the priority order as $t = e_0 \succ e_1 \succ \cdots \succ e_r$ with first priority to the sink $t = e_0$ and priority for intermediate elements as $d_{P_{[s\to e_i]}} > d_{P_{[s\to e_{i+1}]}} \implies e_i \succ e_{i+1}$, for $i = 1, \ldots, r-1$.
(3) Calculate the temporally repeated flow for sink element by using Equation 3.1 and for the the intermediate elements by using Equation 3.2.

Theorem 3.2. *Abstract maximum dynamic flow with intermediate storage obtained by Algorithm 1 using temporally repeated flow is optimal for sink as well as each intermediate elements.*

Proof. First we prove the feasibility of Algorithm 1. Step 1 is to find the shortest distance of each element by using Dijkstra's algorithm and Step 2 is to set priority order of elements, so both steps are feasible. In Step 3, maximum flow at the sink element is obtained as in Ford and Fulkerson [5] but for the intermediate elements, flow is stored using paths $P_{[s\to e]}$, $e \in E_I$ within the time horizon T. So, the flow obtained at each element is feasible.

Next, the optimality of Algorithm 1 is assured by the optimality of Step 3. The optimal flow at sink element is obtained by [5]. Each flow out from the source element that can not reach the sink is stored at either of the intermediate element in priority order. Thus, Algorithm 1 provides the optimal abstract temporally repeated MDF with intermediate storage within the given time horizon T. □

Theorem 3.3. *Algorithm 1 solves Problem 1 in polynomial time.*

Proof. Since the time complexity of Step 1 is $O(|E|^2)$ and by sorting algorithm, complexity of Step 2 is $O(|E|\log(|E|))$. Similarly, temporally repeated formula is applied for each element, so Step 3 can be computed within the time $O(|E|)$. Thus, Algorithm 1 solves an abstract MDF problem with intermediate storage in polynomial time. □

4. Abstract Temporally Repeated MDCF with Intermediate Storage

In this subsection, we discuss the computation of an abstract maximum dynamic contraflow (MDCF) problem with intermediate storage and solve it by using temporally repeated solution. In two way network topology, contraflow means the reversal of the direction of movement capacity of element towards the destination. At the time of disasters, every individual aims to move away from the danger zones towards the safe zones so that paths towards the danger zone are almost empty. Thus, contraflow technique helps to increase the movement capacity of elements by utilizing the unused paths so that maximum amount of flow can be sent within the given time horizon.

Consider an abstract dynamic contraflow network $\mathcal{N} = (E, \overleftrightarrow{\mathcal{P}}, \tau, T)$, where E represents the set of elements and $\overleftrightarrow{\mathcal{P}} = \overrightarrow{P} \cup \overleftarrow{P} \cup \overrightarrow{P}_{[s\to e]} \cup \overleftarrow{P}_{[e\to s]}$ represents the set of two-way paths. Here, $\overrightarrow{P}$ and $\overleftarrow{P}$ represent forward ($P_{[s\to t]}$) and backward ($P_{[t\to s]}$) source-sink paths, respectively. Similarly, we represent $\overrightarrow{P}_{[s\to e]}$ and $\overleftarrow{P}_{[e\to s]}$ for forward and backward intermediate paths, respectively. Our assumption is that the time to transship the flow from an element to its adjacent element in either direction is same. Let $\tau : E \to \mathbb{Z}^+$ be a symmetric transit time between pair of consecutive elements along a path. Then, $\tau_{\overrightarrow{P}_{[e\to a]}} = \tau_{\overleftarrow{P}_{[a\to e]}}$ with $e <_{\overrightarrow{P}} a$ and $a <_{\overleftarrow{P}} e$ so that $\tau_{\overrightarrow{P}} = \tau_{\overleftarrow{P}}$ and $\tau_{\overrightarrow{P}_{[s\to e]}} = \tau_{\overleftarrow{P}_{[e\to s]}}$. The temporal component T represents the time horizon. For static network, transit time τ is considered as cost c and time horizon T is absent. Contrary to general abstract network, incoming movement capacity to the source and outgoing movement capacity from the sink are nonzero for abstract contraflow network. Here, we introduce an abstract MDCF problem with intermediate storage and solve it by using temporally repeated flow.

Problem 2. *For a given abstract dynamic network $\mathcal{N} = (E, \overleftrightarrow{\mathcal{P}}, \tau, T)$, abstract maximum dynamic contraflow problem with intermediate storage is to find the maximum flow leaving the source element that is to be sent to sink via s-t paths $\overrightarrow{P} \cup \overleftarrow{P}$ and allowing the storage of excess flow at intermediate elements e via paths $\overrightarrow{P}_{[s\to e]} \cup \overleftarrow{P}_{[e\to s]}$ $\forall e \in E_I$ with storage capacity $v_e \geq T \sum_{P\in\mathcal{P}: a\in P'} u_a \quad \forall a <_{P'} e,\ \ P' = \overrightarrow{P} \cup \overleftarrow{P}$ within given time horizon T by reverting the direction of paths $\overleftarrow{P}$ and $\overleftarrow{P}_{[e\to s]}$ at time zero.*

To solve the problem, we construct an auxiliary network $\bar{\mathcal{N}} = (E, \bar{\mathcal{P}}, \bar{\tau}, T)$ by adding two-way movement capacities between two consecutive elements. Here, $\bar{\mathcal{P}}$ represent the set of paths in an auxiliary network obtained by reverting the direction of paths $\overleftarrow{P}$ and $\overleftarrow{P}_{[e\to s]}$ at time zero. The movement capacity $\bar{u}_e$ and transit time $\bar{\tau}_e$ are defined as follows: For any two consecutive elements e and a with $e <_{\overrightarrow{P}} a$ and $a <_{\overleftarrow{P}} e$

$$\bar{u}_e = u_{e:e\in\overrightarrow{P}} + u_{a:a\in\overleftarrow{P}}$$

where $u_{a:a\in\overleftarrow{P}} = 0$ if $a \notin \overleftarrow{P}$ and

$$\bar{\tau}_e = \begin{cases} \tau_{e:e\in\overrightarrow{P}} & \text{if } e <_{\overrightarrow{P}} a \\ \tau_{a:a\in\overleftarrow{P}} & \text{otherwise.} \end{cases}$$

We now present an algorithm to solve abstract MDCF problems using temporally repeated flow. We first transform the given two-way network to an auxiliary network $\bar{\mathcal{N}} = (E, \bar{\mathcal{P}}, \bar{\tau}, T)$ and construct cycle free paths on $\bar{\mathcal{N}}$. On these cycle free paths, we solve abstract maximum dynamic flow problem as described in Section 3 by using Algorithm 1 to obtain optimal solution to the corresponding contraflow problem with intermediate storage.

Algorithm 2: Abstract temporally repeated MDCF algorithm with intermediate storage

Input : Given abstract two-way network $\mathcal{N} = (E, \overleftrightarrow{\mathcal{P}}, \tau, T)$.

Output: Abstract temporally repeated MDCF with intermediate storage on $\mathcal{N}$.

(1) Construct an auxiliary network $\bar{\mathcal{N}} = (E, \bar{\mathcal{P}}, \bar{\tau}, T)$.

(2) Construct cycle free paths on $\bar{\mathcal{N}}$ satisfying the switching property.

(3) Compute abstract maximum dynamic flow on $\bar{\mathcal{N}}$ with intermediate storage using Algorithm 1.

(4) A path $\overleftarrow{P}(\overleftarrow{P}_{[e\to s]})$ is reversed if and only if the flow along path $\overrightarrow{P}(\overrightarrow{P}_{[s\to e]})$ is greater than its capacity or if there is a non-negative flow along path $\overrightarrow{P}(\overrightarrow{P}_{[s\to e]}) \notin \overleftrightarrow{\mathcal{P}}$.

Theorem 4.1. *An optimal solution to an abstract MDCF problem with intermediate storage can be obtained from Algorithm 2 by using temporally repeated flow.*

Proof. As each step of Algorithm 2 are feasible, solution obtained from it is feasible. The optimality of Algorithm 2 is dominated by the optimality of Step 3. We compute the abstract MDF in auxiliary network in polynomial time complexity by using Algorithm 1, which is optimal. So Algorithm 2 provides an optimal solution to an abstract MDCF problem with intermediate storage using temporally repeated flow. □

Here, Step 1 of Algorithm 2 can be computed in $O(|E|)$ time and Step 2 can be obtained in polynomial time, [8]. Similarly, Step 3 can be obtained in polynomial time complexity by using Algorithm 1. So the polynomial time solvability of Algorithm 2 is at hand.

Theorem 4.2. *Algorithm 2 solves the abstract MDCF problems with symmetric transit times in polynomial time complexity.*

5. Conclusion

Abstract network permits the flows on paths rather than on arcs satisfying the switching property. In literature, the lexicographically maximum flow, maximum flow over time and earliest arrival flow problems have been solved efficiently in abstract networks. In this paper, we have presented the abstract flow models with intermediate storage in static and dynamic networks. We have solved the MDF problem with intermediate storage by using temporally repeated flow in polynomial time complexity. Similarly, for two way network topology, we have solved abstract

MDCF problem by using temporally repeated flow and present a polynomial time algorithm to solve it . To the best of our knowledge, temporally repeated flow with intermediate storage to solve MDF and MDCF problems in an abstract network topology is introduced for the first time.

Conflict of Interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Data Availability The authors have not used any additional data in this article.

Funding No funding from any source is available for this research.

Acknowledgements The first author (Durga Prasad Khanal) thanks to the German Academic Exchange Service - DAAD for Research Grants - Bi-nationally Supervised Doctoral Degrees/Cotutelle, 2021/22 and University Grants Commission Nepal for PhD Research Fellowship, 2020/21.

References

[1] T.N. Dhamala, S.P. Gupta, D.P. Khanal, U. Pyakurel, Quickest multi-commodity flow over time with partial lane reversals, *Journal of Mathematics and Statistics*, Vol. 16, pp. 198-211, 2020, DOI: https://doi.org/10.3844/jmssp.2020.198.211

[2] E.W. Dijkstra, A note on two problems in connection with graph, *Numer. Math.*, Vol. 1(1), pp. 269-271, 1959.

[3] L. Fleischer, E. Tardos, Efficient continuous-time dynamic network flow algorithms, *Operations Research Letters*, Vol. 23, pp. 71-80, 1998.

[4] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics*, Vol. 8, pp. 399-404, 1956.

[5] L.R. Ford, D.R. Fulkerson, *F*lows in Networks, Princeton University Press, Princeton, New jersey, 1962.

[6] S.P. Gupta, D.P. Khanal, U. Pyakurel, T.N. Dhamala, Approximate algorithms for continuous-time quickest multi-commodity contraflow problem, *The Nepali Mathematical Sciences Report*, Vol. 37, pp. 30-46, 2020, DOI: https://doi.org/10.3126/nmsr.v37i1-2.34068.

[7] A.J. Hoffman, A generalization of max flow - min cut, *Math. Prog.*, Vol. 6, pp. 352-359, 1974.

[8] J.-P.W. Kappmeier, J. Matuschke, B. Peis, Abstract flow over time: A first step towards solving dynamic packing problems, *Theoretical Computer Science, Algorithm and Computation*, Vol. 544, pp. 74-83, 2014.

[9] P.W. Kappmeier, *Generalizations of flows over time with application in evacuation optimization*, PhD Thesis, Technical University, Berlin, Germany, 2015.

[10] D.P. Khanal, U. Pyakurel, T.N. Dhamala, Prioritized multi-commodity flow model and algorithm, *International Symposium on Analytic Hierarchy Process (ISAHP2020)*, 2020, Paper DOI: https://doi.org/10.13033/isahp.y2020.049.

[11] D.P. Khanal, U. Pyakurel, T.N. Dhamala, Maximum multicommodity flow with intermediate storage, *Mathematical Problems in Engineering, Hindawi*, 2021, https://doi.org/10.1155/2021/5063207.

[12] D.P. Khanal, U. Pyakurel, S. Dempe, Dynamic contraflow with orientation dependent transit times allowing intermediate storage, *The Nepali Mathematical Sciences Report*, Vol. 38(2), pp. 1-12, 2021, DOI: http://doi.org/10.3126/nmsr.v38i2.42700.

[13] D.P. Khanal, U. Pyakurel, T.N. Dhamala, S. Dempe, Efficient algorithms for abstract flow with partial switching, *SN Operations Research Forum*, Vol. 3(55), 2022, https://doi.org/10.1007/s43069-022-00168-2.

[14] D.P. Khanal, U. Pyakurel, T.N. Dhamala, S. Dempe, Maximum multi-commodity flow with proportional and flow-dependent capacity sharing, *Comput. Sci. Math. Forum*, Vol. 2(5), 2022, DOI: https://doi.org/10.3390/IOCA2021-10904.

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

[16] M. Martens, *Path-Constrained Network Flows*, PhD Thesis, Technical University, Berlin, Germany, 2007.

[17] M. Martens, S.T. McCormick, A polynomial algorithm for weighted abstract flow, In *Integer Programming and Combinatorial Optimization* in Lecture Notes in Computer Sciences, Vol. 5035, pp. 97-111, 2008.

[18] S.T. McCormick, A polynomial algorithm for abstract maximum flow, In *Proceeding of the* 7^{th} *annual ACM-SIAM symposium on discrete algorithms*, pp. 490-497, 1996.

[19] U. Pyakurel, S. Dempe, Network flow with intermediate storage: models and algorithms, *SN Operations Research Forum*, 2020, DOI:10.1007/s43069-020-00033-0.

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

[21] U. Pyakurel, S. Dempe, Universal maximum flow with intermediate storage for evacuation planning, *In: Kotsireas I.S., Nagurney A., Pardalos P.M., Tsokas A. (eds) Dynamics of Disasters. Springer Optimization and Its Applications*, 169, Springer, Cham. 2021, $https://doi.org/10.1007/978-3-030-64973-9\_14$.

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

[23] U. Pyakurel, S.P. Gupta, D.P. Khanal, T.N. Dhamala, Efficient algorithms on multicommodity flow over time problems with partial lane reversals, *International Journal of Mathematics and Mathematical Sciences, Hindawi*, 2020, DOI: https://doi.org/10.1155/2020/2676378.

[24] U. Pyakurel, D.P. Khanal, T.N. Dhamala, Abstract network flow with intermediate storage for evacuation planning, *European Journal of Operational Research*, 2022, https://doi.org/10.1016/j.ejor.2022.06.054.

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

[26] U. Pyakurel, H.N. Nath, T.N. Dhamala, Partial contraflow with path reversals for evacuation planning, *Annals of Operations Research*, 2019, http://doi.org/10.1007/s10479-018-3031-8.

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