Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5063207, 11 pages
https://doi.org/10.1155/2021/5063207

Research Article

Maximum Multicommodity Flow with Intermediate Storage

Durga Prasad Khanal,[1] Urmila Pyakurel ,[2] and Tanka Nath Dhamala [2]

[1]*Saraswati Multiple Campus, Tribhuvan University, Kathmandu, 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 6 April 2021; Revised 5 June 2021; Accepted 23 June 2021; Published 21 July 2021

Academic Editor: Ioannis T. Christou

The multicommodity flow problem deals with the transshipment of more than one commodity from respective sources to corresponding sinks without violating the capacity constraints. Due to the capacity constraints, flows out from the sources may not reach their sinks, and so, the storage of excess flows at intermediate nodes plays an important role in the maximization of flow values. In this paper, we introduce the maximum static as well as maximum dynamic multicommodity flow problems with intermediate storage. We present polynomial and pseudopolynomial time algorithms for the former and latter problems, respectively. We also present the solution procedures to these problems in contraflow network having symmetric as well as asymmetric arc transit times. We transform the solutions in continuous-time settings by using natural transformation.

1. Introduction

Network is a topological structure with links (arcs), with its crossings (nodes) being its components. The transportation network is one of the relevant examples of a network topology, in which road segments are considered as the arcs and their crossings as nodes. Any kinds of entities moving on the road are considered as flows, and their initial and final destinations are considered as the source and sink, respectively. Each arc has nonnegative capacity, which limits the flow on arc. Dynamic network has one more attribute on arc, i.e., transit time, which represents the time to send the flow from one node to another one. Ford and Fulkerson are the pioneers of the network flow over time (so-called dynamic flow) problems [1, 2].

The transshipment of several different commodities from respective sources to corresponding sinks through a network without violating the capacity constraints on the arcs is known as multicommodity flow problem. Vehicle routine in transportation, production planning, supply chains for essential goods, and massage routing in telecommunication are some examples of multicommodity flow problem. On the basis of temporal dimension, multicommodity flow problem can be classified as static multicommodity flow problem and dynamic multicommodity flow problem [3–6]. If we maximize the supply-demand in a fixed time horizon, then the problem becomes a maximum dynamic multicommodity flow problem. The static multicommodity flow problem is polynomial time solvable by using the ellipsoid or interior point method, whereas dynamic multicommodity flow problem is $\mathcal{NP}$-hard [7]. By using time expanded network, Kappmeier [8] provided the solution of maximum dynamic multicommodity flow problem and multisource single sink multicommodity earliest arrival transshipment problem in pseudopolynomial time complexity. Priority-based multicommodity flow problem and polynomial time solution strategy are presented in [9].

Maximum flow problem with intermediate storage is extremely relevant in large scale disaster management. In evacuation models, one wishes to shift maximum evacuees from danger zones (sources) to safety places (sinks) as quickly and efficiently as possible. Thus, at the time of evacuation, if the number of evacuees out from sources is greater than the minimum cut capacity, then the excess evacuees can be placed at intermediate shelters that are comparatively safer than the danger zone. The various applications of the network flow with intermediate storage are evacuation planning, demand-supply chain of goods, water supply system, etc. Pyakurel and Dempe [10] introduced the concept of maximum static and maximum dynamic flow problems with intermediate storage and presented polynomial time algorithms to solve them. They also presented polynomial time algorithm for dynamic contraflow problem with intermediate storage. In case of multisource multisink network, Pyakurel et al. [11] solved the prioritized maximum flow problem with intermediate storage and presented polynomial time algorithm to solve the problem, where priority is given to the farthest element from the source. Recently, Pyakurel and Dempe [12] presented efficient algorithms for universal maximum dynamic flow problem with intermediate storage in general as well as two-terminal series parallel networks.

In two-way network, contraflow (lane reversal) is one of the best techniques to increase the outbound capacities of arcs and minimize the overall time horizon, in which arcs are reversed towards the destination [13]. Rebennack et al. [14] provided the models and polynomial time algorithms for maximum and quickest flow problems in a two-terminal network by reverting the arcs at time zero and keeping them fixed afterward by using analytical approach for discrete-time settings. In continuous-time settings, Pyakurel and Dhamala [15] introduced the dynamic contraflow model. By using the natural transformation of Fleischer and Tardos [16], they have presented efficient algorithms to solve the maximum, quickest, and earliest arrival flow problems with lane reversals.

Pyakurel et al. [17] 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 proposes like logistic supports and facility locations. Dhamala et al. [18] presented approximation algorithms for quickest multicommodity flow over time problem with partial lane reversals using length bound flow and condensed time expanded network in discrete-time settings. Continuous-time solutions of these problems are found in [19]. Similarly, Pyakurel et al. [20] presented polynomial time algorithm for maximum static and pseudopolynomial algorithm for maximum dynamic multicommodity flow problems with partial lane reversals.

In this paper, we aim to find the solution of discrete-time maximum multicommodity flow problem with intermediate storage by integrating the concept of multicommodity flow problem and the maximum flow problem with intermediate storage. We present polynomial time algorithm for static multicommodity flow problem and pseudopolynomial time algorithm for dynamic multicommodity flow problem by allowing the storage of excess flow at intermediate nodes. We extend the results for contraflow configuration with symmetric as well as asymmetric transit times and also in continuous-time settings.

Our models are designed with the following limitations: at each intermediate node, inflow must be greater or equal to the outflow. At each arc, flow must not exceed the capacity. The storage capacity of intermediate nodes must be at least the sum of incoming arc capacities. Every commodity must transship from respective sources to their corresponding sinks. Objects within a commodity group are homogeneous and between the commodity groups are heterogeneous.

We organize the paper as follows. Section 2 provides the basic terminologies used in the paper and the mathematical formulation of flow models. In Section 3, we present a polynomial time algorithm to solve the maximum static multicommodity flow problem with intermediate storage, and in Section 4, we solve the maximum dynamic multicommodity flow problem with intermediate storage in pseudopolynomial time complexity. For two-way multicommodity network, we present a solution procedure of these problems in Section 5 within the same time complexity. Similarly, in Section 6, we extend the results of dynamic flow problems in continuous-time settings by using natural transformation. The paper is concluded in Section 7.

2. Basic Terminologies and Mathematical Models

Consider a dynamic network $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$, where V and $A \subseteq V \times V$ represent the sets of nodes and arcs with $|V| = n$ and $|A| = m$, respectively. Let $s_i \in S \subset V$ and $t_i \in D \subset V$ be the source and sink nodes with respect to commodity $i \in K = \{1, 2, \ldots, k\}$ and $I = V \setminus \{S, D\}$ the set of intermediate nodes. Here, d_i represents the amount of supply from the source node s_i for each commodity $i \in K$ that is to be sent to the corresponding sink t_i and the intermediate nodes I. Each arc $a = (v, w) \in A$ with head $(a) = w$ and tail $(a) = v$ is equipped with a capacity function $\mathfrak{u}\colon A \longrightarrow \mathcal{R}^+$ that restricts the flow of commodity and a nonnegative transit time function $\tau\colon A \longrightarrow \mathcal{R}^+$ that measures the time to transship the flow from node v to node w. Similarly, $\mathfrak{b}\colon V \longrightarrow \mathcal{R}^+$ represents the storage capacity function of nodes that is used to hold the flow at sources and sinks, together with the storage of excess flow leaving from the source s_i but not reaching the sink t_i at intermediate nodes. Capacity of arcs (roads) and the storage capacity of nodes (shelters) are the controlling parameters of our model, which control the flow at arcs and nodes, respectively. Let $\delta^{\text{out}}(v)$ and $\delta^{\text{in}}(v)$ be the set of outgoing arcs from node v and incoming arcs to node v, respectively. The time period T given in advance is denoted by $\mathcal{T} = \{0, 1, \ldots, T\}$ in discrete-time settings and $\mathcal{T} = [0, T + 1)$ in continuous-time settings. In static flow, the transit time is considered as the cost, and time parameter T is absent.

Throughout the paper, we consider that the storage capacity of sources and sinks is sufficiently large, i.e., $\mathfrak{b}_{s_i} = \mathfrak{b}_{t_i} \leq \infty$ and that of intermediate nodes is finite. If the sum of incoming arc capacities of an intermediate node $v \in I$ is more than the sum of outgoing arc capacities, then the excess flow is used to store at v. Moreover, for the uniqueness of the solution, the storage capacity of $v \in I$ should be $\mathfrak{b}_v \geq \sum_{a \in \delta^{\text{in}}(v)} \mathfrak{u}_a$.

2.1. Static Multicommodity Flow Model. The static multicommodity flow function g on the given network $\mathcal{N} = (V, A, K, c, \mathfrak{u}, \mathfrak{b}, d_i, S, D)$ is the sum of nonnegative arc flow functions $g_a^i\colon A \longrightarrow \mathcal{R}^+$ and the excess flow functions

g_v^i: $I \longrightarrow \mathscr{R}^+$, for each $i \in K$, satisfying conditions (1)–(5). The linear programming formulation of static multi-commodity flow with intermediate storage is as follows:

$$\max d_i = \sum_{a\in\delta^{\text{out}}(s_i)} g_a^i = \sum_{a\in\delta^{\text{in}}(t_i)} g_a^i + \sum_{v\in I:\ \mathfrak{b}_v\geq 0} g_v^i, \quad (1)$$

such that

$$\sum_{a\in\delta^{\text{in}}(v)} g_a^i - \sum_{a\in\delta^{\text{out}}(v)} g_a^i \geq 0, \quad \forall v \in I, i \in K, \quad (2)$$

$$0 \leq g_a = \sum_{i\in K} g_a^i \leq \mathfrak{u}_a, \quad \forall a \in A, \quad (3)$$

$$0 \leq g_v = \sum_{i\in K} g_v^i \leq \mathfrak{b}_v, \quad \forall v \in I, \quad (4)$$

$$\sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a \leq \mathfrak{b}_v, \quad \forall v \in I.pt \quad (5)$$

Objective function in equation (1) is to maximize the total flow out from each source, for all $i \in K$, which is equal to the sum of inflow at the sink and the excess flow at intermediate nodes. Equation (2) represents the nonconservation of flow at intermediate nodes. The constraint in (3) represents the bundle constraint on each arc that is bounded by its capacity, and the constraints in (4) represent the excess flow at each intermediate node, which is bounded by the storage capacity. Similarly, the constraint in (5) represents that the storage capacity of intermediate node $v \in I$ is at least the sum of incoming arc capacities to v. The cost of static flow g associated with arc a and commodity i with cost coefficient c_a^i is defined as

$$c(g) = \sum_{i\in K}\sum_{a\in A} c_a^i g_a^i. \quad (6)$$

2.2. Dynamic Multicommodity Flow Model. For a given dynamic network $\mathscr{N}$ with constant transit time τ on each arc a, the multicommodity flow over time function ψ is the sum of nonnegative arc flow functions ψ^i: $A \times \mathscr{T} \longrightarrow \mathscr{R}^+$ and the storage flow functions ψ_v^i: $I \times \mathscr{T} \longrightarrow \mathscr{R}^+$ for each $i \in K$, satisfying constraints (7)–(11). The linear programming formulation of dynamic multicommodity flow with intermediate storage is as follows:

$$\max d_i = \sum_{a\in\delta^{\text{out}}(s_i)} \sum_{\theta=0}^{T} \psi_a^i(\theta) = \sum_{a\in\delta^{in}(t_i)} \sum_{\theta=\tau_a}^{T} \psi_a^i(\theta - \tau_a) + \sum_{v\in I:\ \mathfrak{b}_v\geq 0} \psi_v^i(T), \quad (7)$$

such that

$$\sum_{a\in\delta^{\text{in}}(v)} \sum_{\beta=\tau_a}^{\theta} \psi_a^i(\beta - \tau_a) - \sum_{a\in\delta^{\text{out}}(v)} \sum_{\beta=0}^{\theta} \psi_a^i(\beta) \geq 0, \quad \forall v \in I, i \in K, \theta \in \mathscr{T}, \quad (8)$$

$$0 \leq \psi_a(\theta) = \sum_{i\in K} \psi_a^i(\theta) \leq \mathfrak{u}_a, \quad \forall a \in A, \theta \in \mathscr{T}, \quad (9)$$

$$0 \leq \psi_v(\theta) = \sum_{i\in K} \psi_v^i(\theta) \leq \mathfrak{b}_v, \quad \forall v \in I, \theta \in \mathscr{T}, \quad (10)$$

$$\sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a \leq \mathfrak{b}_v \leq T \sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a, \quad \forall v \in I. \quad (11)$$

Equation (7) is an objective function that maximizes the total flow out from the source in time horizon T, for each $i \in K$, which is equal to the sum of inflow at sink and the excess flow at intermediate nodes. Equation (8) represents the nonconservation of flow at intermediate nodes for each time step θ. In any instance of time θ, the bundle constraint in (9) is bounded by arc capacity, and the constraint in (10) represents that the excess flow at each intermediate node is bounded by the storage capacity. Similarly, the constraint in (11) represents the lower and upper bounds of the storage capacity of intermediate node $v \in I$. The cost of discrete dynamic flow ψ associated with arc a and commodity i with cost coefficient c_a^i is defined as

$$c(\psi) = \sum_{i\in K}\sum_{a\in A} c_a^i \sum_{\theta=0}^{T} \psi_a^i(\theta). \quad (12)$$

3. Maximum Static Multicommodity Flow

In this section, we introduce the maximum static multicommodity flow problem with intermediate storage and present a polynomial time algorithm to solve it.

Problem 1. For a given static network $\mathscr{N} = (V, A, K, c, \mathfrak{u}, \mathfrak{b}, d_i, S, D)$, the maximum static multicommodity flow problem with intermediate storage finds the maximization of flow leaving from each source s_i, for all $i \in K$, which is to be sent to the corresponding sink t_i via $s_i - t_i$ paths by allowing the storage of maximum excess flow at intermediate nodes with storage capacity $\mathfrak{b}_v \geq \sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a$.

As the solution strategy, we first reduce the multicommodity flow problem into k independent single commodity flow problems by reallocating the capacity of bundle arcs using the resource directive decomposition method. It reallocates the capacity of bundle arc for each commodity in such a way that the objective is optimal. The decomposition algorithm to minimal-cost multicommodity flow problem can be used for minimum cost flow problem, which was the motivation for the development of the original Dantzig–Wolfe decomposition method [21] (see Bazaraa et al. [22]). For each $i \in K$, we used to store the maximum flow at sink t_i and the excess flow at intermediate nodes $v \in I$ with priority order. As in Pyakurel and Dempe [10], we have a single sink for each commodity $i \in K$, which is considered as the most appropriate place to store the flow. So, the first priority is given to the sink to transship as much flow as possible. To store the excess flow at intermediate nodes, we set the priority order as follows: for each $v \in I$ with storage capacity $\mathfrak{b}_v \geq \sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a$, calculate the shortest distance $d_{[s_i,v]}$, for each i, by using algorithm of Dijkstra [23]. We consider the path with the minimum cost as the shortest path, and the priority is given to the farthest node among the nodes with shortest distance. That is, $\forall v_1, v_2 \in I$ if $d_{[s_i,v_1]} > d_{[s_i,v_2]}$, then v_1 is higher in priority than v_2 and it is denoted by $v_1 \succ v_2$. It is to be noted that the nodes lying in the bundle arcs may have different priority ordering with respect to the commodity.

For each prioritized node $v \in I$, we create dummy port v_i' (since the node $v \in I$ lying in the bundle arc contains the flow of more than one commodity, so dummy ports are represented commodity-wise, i.e., v_i') with cost $c_{[v,v_i']} = d_{[v,v_i']} = 0$ and capacity $\mathfrak{u}_{[v,v_i']} = \mathfrak{b}_v = \mathfrak{b}_{v_i'}$, where $\mathfrak{u}_{[v,v_i']}$ and $c_{[v,v_i']}$ are the arc capacity and cost of dummy arc (v, v_i'), respectively. Every dummy port v_i' with respect to commodity i has the same priority order as v has. Associated with each commodity i, the collection of dummy ports $\{v_i'\}$ together with the sink t_i forms a modified network $\mathcal{N}_i' = (V_i', A_i', c, \mathfrak{u}, \mathfrak{b}, d_i, s_i, D_i')$ with single source s_i and multiple sink $D_i' = \{t_i \cup \{v_i'\}\}$. For an instance, if $t_i \succ v_{i,1} \succ \ldots \succ v_{i,r}$ is priority order of nodes with respect to commodity i, then $D_i' = \{t_i = v_{i,0}', v_{i,1}', \ldots, v_{i,r}'\}$.

Now, we present a polynomial time algorithm to solve Problem 1 by using the algorithm of Pyakurel and Dempe [10] for single source multisink network $\mathcal{N}_i'$, $\forall i \in K$.

Theorem 1. *Algorithm 1 solves the maximum static multicommodity flow problem with intermediate storage optimally.*

Proof. Before proving the optimality, we first prove the feasibility of the algorithm. Step 1 is the use of decomposition algorithm to reduce the multicommodity flow problem to single commodity flow problem, and Step 2 calculates the shortest distances by using Dijkstra's algorithm, so both steps are feasible. Steps 3, 4, and 6 are prioritization of nodes, modification of network, and transformation of solution, which can be solved in linear time, and so they are feasible. Similarly, according to Pyakurel and Dempe [10], Step 5 provides feasible flow with intermediate storage for each commodity $i \in K$. Thus, the solution obtained from Algorithm 1 is feasible.

The optimality of algorithm is assured by Step 5. For each commodity $i \in K$, lexicographic maximum static flow in prioritized sink D_i' is obtained optimally as the single commodity flow problem solved by Pyakurel and Dempe [10]. So, the sum of optimal single commodity flows $\sum_i d_i$ is optimal multicommodity flow with intermediate storage. □

Theorem 2. *Maximum static multicommodity flow problem with intermediate storage can be solved in polynomial time complexity by using Algorithm 1.*

Proof. The decomposition algorithm in Step 1 is obtained in polynomial time complexity, and the shortest distance can be obtained in $O(n^2)$ time by using Dijkstra's algorithm. The prioritization of nodes and creating dummy ports can be obtained in linear time. For each single commodity i, Step 5 can be solved in polynomial time complexity of $O(\delta mn)$ for $0 < \delta < n$, [10]. Therefore, Algorithm 1 solves the maximum static multicommodity flow problem with intermediate storage in polynomial time with complexity $O(n^2) + O(k\delta mn)$, where $|K| = k$. □

Example 1. Consider a two-commodity network with capacity and cost on each arc as shown in Figure 1(a), where the numbers aside the nodes represent the node capacities. Using Dijkstra's algorithm, we find the shortest distance of each intermediate node and fix the priority order with farther-in-distance-higher-in-priority. So, the priority orders are $t_1 \succ y \succ x$ and $t_2 \succ x \succ y$ for commodity 1 and commodity 2, respectively. After priority ordering, we denote $D_1' = \{t_1, y_1', x_1'\}$ and $D_2' = \{t_2, x_2', y_2'\}$ as the set of prioritized dummy ports for commodity 1 and commodity 2, respectively, which is presented in Figure 1(b).

While decomposing the flow on the bundle arc (x, y), flows of 3 and 2 units are assigned for commodity 1 and commodity 2, respectively. By using Algorithm 1, the maximum amount of flow leaving the source s_1 is 4 units out of which 2 units of flow are reached at sink t_1 and the intermediate nodes y and x hold 1 unit each. Similarly, 7 units of flow are transmitted from the source s_2, out of which 6 units of flow are reached at the sink t_2 and the intermediate nodes x and y hold 1 and 0 units, respectively. At last, the solution is transformed to the original network by removing the dummy ports and dummy arcs and sending back the flow to its respective nodes. The amount of flow stored at sinks and the intermediate nodes is $g_{t_1} = 2$, $g_{t_2} = 6$, $g_x = 2$, and $g_y = 1$. If the intermediate storage is not permitted, then the flow of $g_{t_1} = 2$ and $g_{t_2} = 6$ units can only be transshipped.

4. Maximum Dynamic Multicommodity Flow

This section deals with the maximum dynamic multicommodity flow problem, where storage of the excess flow at intermediate nodes is allowed. We present a pseudopolynomial time algorithm based on the time expanded network of Kappmeier [8] to solve the problem.

Problem 2. For a given dynamic network $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$, the maximum dynamic multicommodity flow problem with intermediate storage is to maximize the flow leaving each source s_i, for all $i \in K$, which is to be sent to the corresponding sink t_i via $s_i - t_i$ paths by allowing the storage of maximum excess flow at the intermediate nodes with storage capacity $\sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a \leq \mathfrak{b}_v \leq T\sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a$ within time horizon T.

```
Input: given static network $\mathcal{N} = (V, A, K, c, \mathfrak{u}, \mathfrak{b}, d_i, S, D)$.
Output: maximum static multicommodity flow with intermediate storage in $\mathcal{N}$.
(1) Reconfigure the multicommodity flow problem into $k$ independent single commodity flow problems by reallocating the capacity of
    bundle arcs using the resource directive decomposition.
(2) For each $v \in I$ with $\mathfrak{b}_v \geq \sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a$, compute commodity-wise shortest distance $d_{[s_i,v]}$ by using Dijkstra's algorithm.
(3) Fix the priority order as $t_i \succ v_{i,1} \succ \dots \succ v_{i,r}$ with respect to commodity $i$ with first priority to the sink $t_i$ and priority for intermediate
    elements as $d_{[s_i,v_{i,m}]} > d_{[s_i,v_{i,m+1}]} \Rightarrow v_{i,m} \succ v_{i,m+1}$, for $m = 1, \dots, r-1$.
(4) For each $i \in K$, construct the modified network $\mathcal{N}'_i = (V'_i, A'_i, c, \mathfrak{u}, \mathfrak{b}, d_i, s_i, D'_i)$ with single source $s_i$ and multiple sinks with dummy
    ports $D'_i = \{t_i = v'_{i,0}, v'_{i,1}, \dots, v'_{i,r}\}$.
(5) For $i = 1, \dots, k$:
        Compute the lexicographic maximum static flow in $\mathcal{N}'_i$ with priority order of Step 3 according to [10].
(6) Transform the solution to the original network $\mathcal{N}$ by removing the dummy ports and the dummy arcs.
```

Algorithm 1: Maximum static multicommodity flow algorithm with intermediate storage (MSMCFAIS).

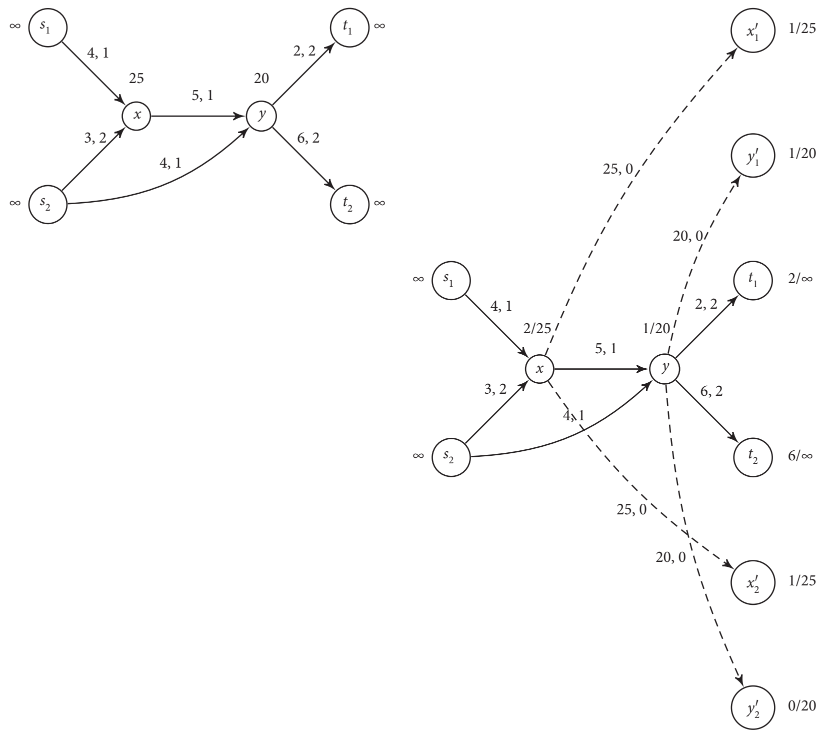

Figure 1: (a) A two-commodity network with capacity and cost on the arcs and storage capacity on the nodes. (b) The modified network of (a) with prioritized dummy ports.

As in Section 3, we first reduce the multicommodity flow problem into k independent subproblems and fix the priority order of intermediate nodes. Static solution is obtained in the modified single source and multisink network $\mathcal{N}'_i = (V'_i, A'_i, \mathfrak{u}, \mathfrak{b}, \tau, d_i, s_i, D'_i, T)$ for all $i \in K$, by using Algorithm 1. To obtain the dynamic solution, we use the static multicommodity flow on time expanded network as in Kappmeier [8].

For this, we construct a temporary sink $\bar{t}_i$ with infinite capacity and join each dummy port $v'_i \in D'_i$ with $\mathfrak{u}_{[v'_i,\bar{t}_i]} = g^i_{v'_i}(\theta)$ and $\tau_{[v'_i,\bar{t}_i]} = 0$, where $g^i_{v'_i}(\theta)$ is the static flow of commodity i at v'_i at time θ. The choice of $\mathfrak{u}_{[v'_i,\bar{t}_i]} = g^i_{v'_i}(\theta)$ is taken to assure that the flow while sending back from the dummy ports must be on respective nodes. Now, for each $i \in K$, the new network $\widetilde{\mathcal{N}}'_i$ is obtained by adding temporary sink and arcs in $\mathcal{N}'_i$ so that it becomes a single source single sink network with prioritized intermediate nodes. Let $\mathcal{N}^T = (V_T, A_T = A_M \cup A_H \cup A^s \cup A^t, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, \widehat{S}, \widehat{D}, T)$ be the time expanded network of new network $\widetilde{\mathcal{N}}' = \cup_i \widetilde{\mathcal{N}}'_i$, where $\widetilde{\mathcal{N}}'_i$ is obtained by including temporary sinks, supersource, and supersink in $\mathcal{N}'_i$. The components in time expanded network $\mathcal{N}^T$ are defined as follows:

$$\begin{aligned}
V_T &= \{v_\theta\colon v \in V, \theta \in \mathcal{T}\} \cup \left\{\{v_{i,\theta}'\} \cup \{\bar{t}_{i,\theta}\}\colon i \in K, \theta \in \mathcal{T}\right\} \cup \{s_i^*, \bar{t}_i^*\colon i \in K\} \cup \{\tilde{s}, \tilde{t}\},\\
A_M &= \left\{(v_\theta, w_{\theta+\tau_a})\colon a = (v,w) \in A, \theta \in \mathcal{T}\right\} \cup \left\{(v_\theta, v_{i,\theta}') \cup (v_{i,\theta}', \bar{t}_{i,\theta})\colon v \in I, \theta \in \mathcal{T}\right\},\\
A_H &= \{(v_\theta, v_{\theta+1})\colon v \in V, \theta \in \mathcal{T}\} \cup \left\{(v_{i,\theta}', v_{i,\theta+1}') \cup (\bar{t}_{i,\theta}, \bar{t}_{i,\theta+1})\colon i \in K, \theta \in \mathcal{T}\right\},\\
A^s &= \{(\tilde{s}, s_i^*)\colon i \in K\} \cup \left\{(s_i^*, s_{i,\theta})\colon i \in K, \theta \in \mathcal{T}\right\},\\
A^t &= \{(\tilde{t}, \bar{t}_i^*)\colon i \in K\} \cup \left\{(\bar{t}_i^*, \bar{t}_{i,\theta})\colon i \in K, \theta \in \mathcal{T}\right\},\\
\widehat{S} &= \left\{\tilde{s}, s_i^*, s_{i,\theta}\colon i \in K, \theta \in \mathcal{T}\right\},\\
\widehat{D} &= \left\{\tilde{t}, t_i^*, \bar{t}_{i,\theta}\colon i \in K, \theta \in \mathcal{T}\right\}.
\end{aligned} \tag{13}$$

Kappmeier [8] has shown that the static multicommodity flow on the time expanded network is equivalent to the dynamic multicommodity flow on the original network.

Theorem 3. *(see [8]). For a given dynamic network $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$ with time horizon T, a feasible static $\widehat{S} - \widehat{D}$ multicommodity flow g^T in the time expanded network $\mathcal{N}^T$ yields the feasible dynamic multicommodity flow ψ in $\mathcal{N}$ such that $|g^T| = |\psi|$.*

We now present an algorithm to solve problem 2 by using time expanded network.

Theorem 4. *Algorithm 2 provides an optimal solution to the maximum dynamic multicommodity flow problem with intermediate storage in pseudopolynomial time complexity.*

Proof. At first, we prove the feasibility of Algorithm 2. As Step 1 is a reconfiguration of given network by using decomposition algorithm, and Step 2 is its modification including dummy ports and temporary sinks, so these steps provide the feasible solution. Due to Theorem 3, construction of time expanded network in Step 3 is feasible, and the feasibility of Step 4 is obtained by Algorithm 1. The transformation of solution in original network is also feasible. The optimality of algorithm is assured by the optimality of Step 4, which is as similar to [8].

Next, we prove the computational time of Algorithm 2, which is dominated by the complexity of Step 4. Here, the computational time complexity of Step 4 is $O(T \times \text{MSMCFAIS})$, where $O(\text{MSMCFAIS})$ is the complexity of maximum static multicommodity flow algorithm with intermediate storage. From Theorem 2, $O(\text{MSMCFAIS}) = O(n^2) + O(k\delta mn)$ for $0 < \delta < n$. With Step 4 being polynomial in input size of the network, but not bounded in T, Algorithm 2 solves the maximum dynamic multicommodity flow problem with intermediate storage in pseudopolynomial time. □

Example 2. Consider a two-commodity network from Figure 1 by considering the cost on each arc as the transit time with time horizon $T = 5$. The prioritization of intermediate nodes and creating dummy ports are similar to Example 1. For each commodity $i = 1, 2$, the problem is reduced to single source and multisink single commodity flow problem due to dummy ports (see Figure 1(b)). By adding temporary sink $\bar{t}_i$, it reduces to commodity-wise single source single sink (i.e., $s_i - \bar{t}_i$) problem, which is shown in Figure 2. We calculate the maximum static multicommodity flow with intermediate storage using Algorithm 1 and then repeat the procedure with intermediate storage in time expanded network as similar to Kappmeier [8]. Here, the maximum static flow is calculated from minimum cost flow by considering the transit time as cost. Figure 3 represents the time expanded network of Figure 2 with time horizon $T = 5$. At last, we obtain the maximum dynamic flow with intermediate storage by removing the dummy arcs and replacing the flow of dummy ports to their respective nodes.

For commodity 1, total amount of flow leaving the source s_1 within the time horizon $T = 5$ along the path $s_1 - x - y - t_1$ is 20 units, out of which 4, 8, and 8 units are transshipped with priority order at t_1, y and x, respectively. For commodity 2, flows are sent through two paths $s_2 - x - y - t_2$ and $s_2 - y - t_2$ with priority order $t_2 \succ x \succ y$. It is to be noted that while sending flow from the path $s_2 - x - y - t_2$, flow leaving s_2 at $\theta = 0$ sends 2 units of flow at t_2 by storing 1 unit at x. After next iteration onward, flow cannot reach the sink, and so it is to be stored at x but not at y because x is higher in priority than y. Similarly, path $s_2 - y - t_2$ first sends 4 units of flow thrice at t_2 and then holds the flow at y for next two times. Total amount of flow leaving the source s_2 through two paths within $T = 5$ is 32 units, out of which 14, 10, and 8 units are transshipped at t_2, x and y, respectively. The detailed information of the flow leaving from sources at different time steps θ, which are to be stored at sinks and the intermediate nodes, is presented in Table 1.

Here, the amount of flow reaching the sinks is the maximum multicommodity flow without intermediate storage. So, if intermediate storage is prohibited, then only 4 units of flow of commodity 1 and 14 units of flow of commodity 2 can be reached at their respective sinks within the time $T = 5$.

5. Multicommodity Contraflow Problems

In this section, we investigate the multicommodity flow problem with contraflow configuration, where the storage of excess flow at intermediate nodes is allowed. In a two-way network, contraflow means the reversal of oppositely directed arcs towards the destination node to improve the flow and reduce the overall

Input: given dynamic network $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$.
Output: maximum dynamic multicommodity flow with intermediate storage in $\mathcal{N}$.

(1) Reconfigure the multicommodity flow problem into k independent single commodity flow problems by reallocating the capacity of bundle arcs by using the resource directive decomposition and considering the transit times as cost.
(2) Construct a modified new network $\widetilde{\mathcal{N}}'$ by including temporary sinks, supersource, and supersink in $\mathcal{N}_i'$.
(3) Using new network $\widetilde{\mathcal{N}}'$, construct the time expanded network $\mathcal{N}^T = (V_T, A_T = A_M \cup A_H \cup A^s \cup A^t, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, \widehat{S}, \widehat{D}, T)$.
(4) Calculate the maximum static multicommodity flow with intermediate storage on the time expanded network $\mathcal{N}^T$ by using Algorithm 1.
(5) Transform the solution to the original network $\mathcal{N}$ by removing the dummy ports, temporary sinks, supersource, supersink, and dummy arcs.

Algorithm 2: Maximum dynamic multicommodity flow algorithm with intermediate storage (MDMCFAIS).

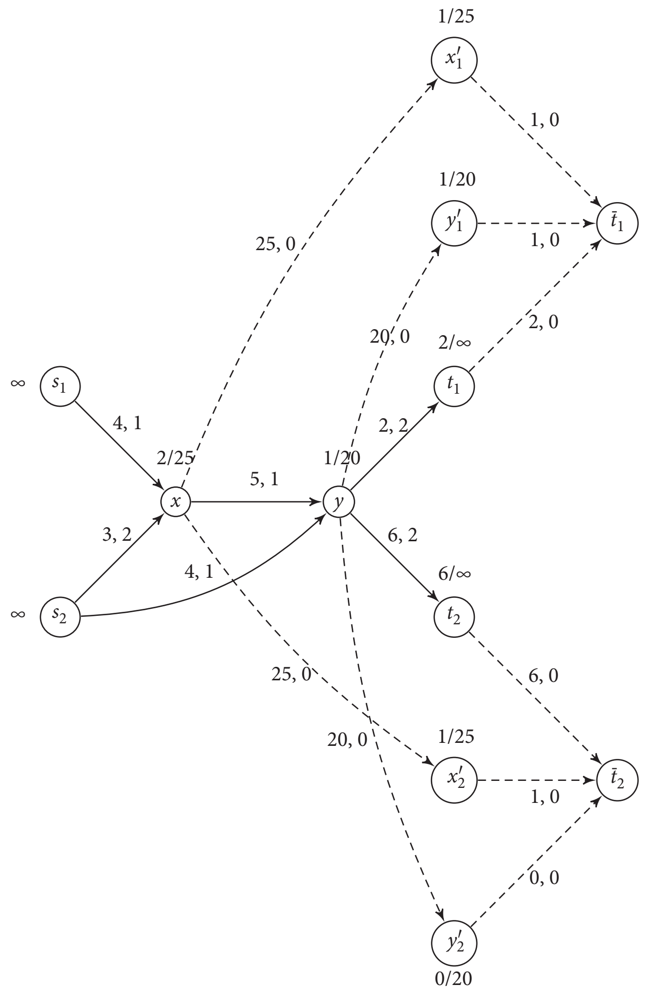

Figure 2: Modified new network $\widetilde{\mathcal{N}}'$ with dummy ports and temporary sinks.

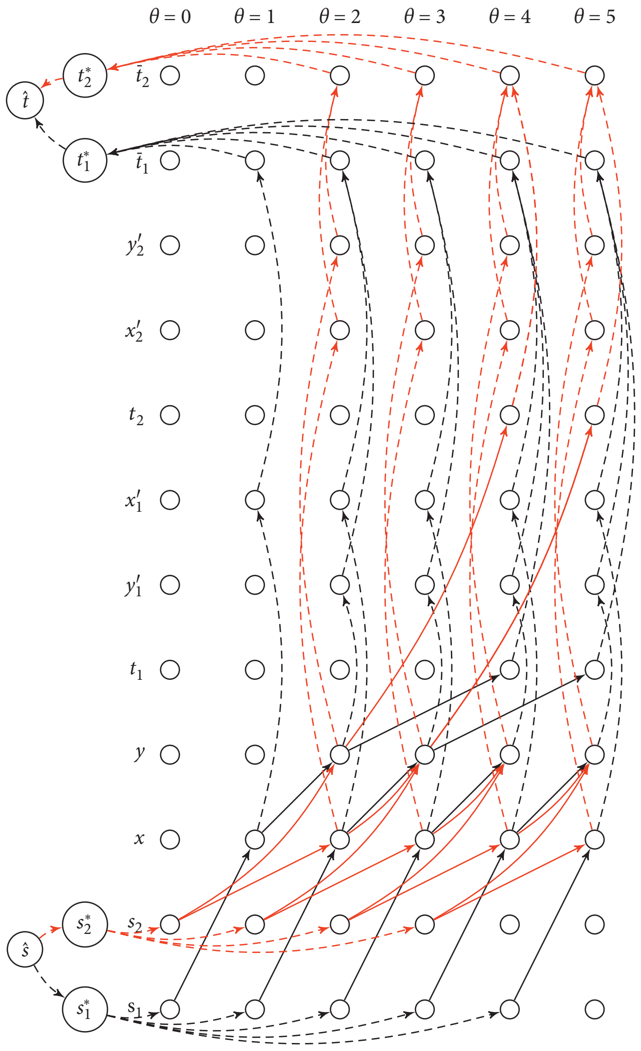

Figure 3: Time expanded multicommodity network flow with intermediate storage, where black and red colors are for commodity 1 and commodity 2, respectively. Dotted arcs represent the dummy arcs.

time horizon. We discuss two different aspects of transit times (or cost for static), symmetric and asymmetric, between the pair of nodes with oppositely directed arcs.

5.1. Contraflow with Symmetric Transit Times. Let $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$ be a dynamic multicommodity network having symmetric transit times on antiparallel arcs, i.e., $\tau_a = \tau_{\overleftarrow{a}}$, where $\overleftarrow{a} = (w, v)$ is oppositely directed arc of $a = (v, w)$. To solve the problem in contraflow network, we transform the given network to an auxiliary network as follows.

Table 1: Multicommodity flow with intermediate storage in each time θ.

	Commodity 1					Commodity 2			
	Path: $s_1 - x - y - t_1$					Path: $s_2 - x - y - t_2$			
	Flow at					Flow at			
Start time at s_1	x	y	t_1	Reaching time at last node	Start time at s_2	x	y	t_2	Reaching time at last node
$\theta = 0$	1	1	2	$\theta = 4$ at t_1	$\theta = 0$	1	0	2	$\theta = 5$ at t_2
$\theta = 1$	1	1	2	$\theta = 5$ at t_1	$\theta = 1$	3	×	×	$\theta = 3$ at x
$\theta = 2$	1	3	×	$\theta = 4$ at y	$\theta = 2$	3	×	×	$\theta = 4$ at x
$\theta = 3$	1	3	×	$\theta = 5$ at y	$\theta = 3$	3	×	×	$\theta = 5$ at x
$\theta = 4$	4	×	×	$\theta = 5$ at x		Path: $s_2 - y - t_2$			
					$\theta = 0$	×	0	4	$\theta = 3$ at t_2
					$\theta = 1$	×	0	4	$\theta = 4$ at t_2
					$\theta = 2$	×	0	4	$\theta = 5$ at t_2
					$\theta = 3$	×	4	×	$\theta = 4$ at y
					$\theta = 4$	×	4	×	$\theta = 5$ at y
Total	8	8	4	Total $d_1 = 20$	Total	10	8	14	Total $d_2 = 32$

For a given two-way multicommodity network $\mathcal{N}$ with symmetric transit times, the corresponding auxiliary network is denoted by $\overline{\mathcal{N}}$ with network topology $\overline{\mathcal{N}} = (V, \overline{A}, K, \overline{\mathfrak{u}}, \mathfrak{b}, \overline{\tau}, d_i, S, D, T)$, containing undirected edges $\overline{A} = \{(v, w) \colon (v, w) \text{ or } (w, v) \in A\}$. The capacity of an arc in an auxiliary network is the sum of capacities of arcs a and $\overleftarrow{a}$ such that $\overline{\mathfrak{u}}_a = \mathfrak{u}_a + \mathfrak{u}_{\overleftarrow{a}}$, where $\mathfrak{u}_a = 0$ if $a \notin A$. The transit time of an arc in an auxiliary network is

$$\overline{\tau}_a = \begin{cases} \tau_a, & \text{if } a \in A, \\ \tau_{\overleftarrow{a}}, & \text{otherwise.} \end{cases} \tag{14}$$

All other parameters are the same as in $\mathcal{N}$. Contrary to the general network, incoming arcs to the sources s_i and outgoing arcs from the sinks t_i may be present in the contraflow network for all $i \in K$.

We now present the maximum dynamic contraflow problem with intermediate storage herein.

Problem 3. For a given dynamic network $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$, the maximum dynamic multicommodity contraflow problem with intermediate storage is to find the maximum flow leaving from each source s_i, $\forall i \in K$, which is to be sent to their respective sinks t_i via $s_i - t_i$ paths by allowing the storage of maximum excess flow at intermediate nodes $v \in I$ with storage capacity $\sum_{a \in \delta^{\text{in}}(v)} \mathfrak{u}_a \leq \mathfrak{b}_v \leq T \sum_{a \in \delta^{\text{in}}(v)} \mathfrak{u}_a$, $\forall a \in \overline{A}$ within the given time horizon T by reverting the direction of arcs at time zero.

To solve the problem, we present an algorithm based on the time expanded network of an auxiliary network $\overline{\mathcal{N}}$ as follows.

We first transform the given two-way network into an auxiliary network $\overline{\mathcal{N}}$. As in Section 3, we decompose the multicommodity flow problem to k single commodity flow problems and then fix the priority order of each intermediate node. On each cycle free path of auxiliary network $\overline{\mathcal{N}}$, we solve the maximum dynamic multicommodity flow problem, for each $i \in K$, as described in Section 4.

Here, Steps 1, 3, and 4 of Algorithm 3 can be computed in $O(E)$ time. In Step 2, we can find a pseudopolynomial time solution for dynamic multicommodity contraflow problem with intermediate storage in $\overline{\mathcal{N}}$ (as in Theorem 4). So, the maximum dynamic multicommodity contraflow problem with intermediate storage can be solved in pseudopolynomial time by using Algorithm 3.

Theorem 5. *Algorithm 3 solves the maximum dynamic multicommodity contraflow problem with intermediate storage in pseudopolynomial time complexity.*

Remark 1. If we consider the cost instead of transit time and apply Algorithm 1 in Step 2 of Algorithm 3, then the solution of maximum static multicommodity contraflow problem with intermediate storage can be obtained in polynomial time complexity.

5.2. Contraflow with Orientation-Dependent Transit Times.

If the transit times on antiparallel arcs of a two-way network are not identical, then it is known as the network with asymmetric transit times. For a network with asymmetric transit times, if the transit times of arcs in an auxiliary network are taken as the orientation of the arcs, then it is known as orientation-dependent transit time. Nath et al. [24] considered the orientation-dependent asymmetric transit times of reversed lanes in general form and presented strongly polynomial time algorithms to solve the single source single sink maximum dynamic and quickest contraflow problems. Here, we discuss about the multicommodity contraflow problem with intermediate storage by taking orientation-dependent transit times.

Let $\mathcal{N} = (V, A, K, \mathfrak{u}, \mathfrak{b}, \tau, d_i, S, D, T)$ be a two-way dynamic network with asymmetric nonzero transit times τ on arcs, so that $\tau_a \neq \tau_{\overleftarrow{a}}$. We construct an auxiliary network $\overline{\mathcal{N}} = (V, \overline{A}, K, \overline{\mathfrak{u}}, \mathfrak{b}, \overline{\tau}, d_i, S, D, T)$, where $\overline{A}$ is obtained by reverting the direction of arcs $\overleftarrow{a}$ at time zero. The arc capacity $\overline{\mathfrak{u}}$ and transit time $\overline{\tau}_a$ can be obtained as follows:

$$\overline{\mathfrak{u}}_a = \mathfrak{u}_a + \mathfrak{u}_{\overleftarrow{a}}, \tag{15}$$

```
**Input:** given two-way dynamic multicommodity network $\mathcal{N}$.
**Output:** maximum dynamic multicommodity contraflow with intermediate storage.
(1) Construct an auxiliary network $\overline{\mathcal{N}}$ of $\mathcal{N}$.
(2) Compute the maximum dynamic multicommodity flow with intermediate storage in $\overline{\mathcal{N}}$ by using Algorithm 2.
(3) Decompose the flow along $s_i - t_i$ paths and cycles, and remove the flows in cycles $\forall i \in K$.
(4) An arc $\overleftarrow{a}$ is reversed if and only if the flow along arc $a$ is greater than its capacity, or if there is a nonnegative flow along arc $a \notin A$.
```

Algorithm 3: Maximum dynamic multicommodity contraflow algorithm with intermediate storage.

where $\mathfrak{u}_a = 0$ if $a \notin A$ and

$$\overline{\tau}_a = \begin{cases} \tau_a \text{ if the orientation of arc is along a,} \\ \tau_{\overleftarrow{a}} \text{ if the orientation of arc is along } \overleftarrow{a}. \end{cases} \quad (16)$$

By using Algorithm 3, the optimal solution for maximum static and maximum dynamic multicommodity contraflow problems with intermediate storage can be obtained for a given network with asymmetric transit times, where transit time is considered as a cost in static problem.

If, on the other hand, the transit time attributes on the evacuation network are antisymmetric, but one wishes to adopt contraflow approach with the same transit time on the reversed arcs as it was before its reversal, Algorithm 3 obviously works well as that can be interpreted as symmetric transit time solution. It can be applied if two nodes in a network have parallel connections with different transit times.

6. Continuous Dynamic Multicommodity Flow

We discussed the maximum dynamic flow and contraflow problems with intermediate storage in Section 4 and Section 5, respectively, where transit times are taken in discrete settings. Actually, discrete dynamic flow function ψ assigns the flow from the source node at each time step $\theta = 0, \ldots, T$ satisfying the capacity constraints. In this section, we formulate the dynamic multicommodity flow with intermediate storage in continuous-time settings. A continuous dynamic flow function ψ^c with intermediate storage is defined as the flow rate per unit time that leaves from the source at each moment of time by allowing the storage of excess flow at intermediate nodes without violating the capacity constraints.

By using natural transformation, Fleischer and Tardos [16] established the relation between discrete and continuous flow models. This natural transformation defines the continuous dynamic flow for time interval $[\theta, \theta+1)$ with $\psi^c_a[\theta, \theta+1) = \psi_a(\theta)$, where $\psi_a(\theta)$ is the amount of discrete dynamic flow entering arc $a \in A$ at each time step $\theta = 0, \ldots, T$. Here, we use this logic to transform the discrete-time multicommodity flow to continuous-time multicommodity flow with intermediate storage as follows: any discrete multicommodity flow over time ψ^i_a with integral time horizon T is equivalent to the continuous multicommodity flow over time $\psi^{i,c}_a[\theta, \theta+1)$ by incorporating the flow ψ entering an arc a at time step $\theta \le T - \tau_a$ as a constant flow rate on arc a during the unit time interval $[\theta, \theta+1)$ by allowing the storage of excess flow at intermediate nodes. Mathematically,

$$\int_{\theta}^{\theta+1} \psi^{i,c}_a(\alpha)\mathrm{d}\alpha = \psi^i_a(\theta), \quad \forall a \in A, i \in K. \quad (17)$$

We formulate the dynamic multicommodity flow models with intermediate storage in continuous settings as follows:

$$\max d_i = \sum_{a\in\delta^{\text{out}}(s_i)} \int_{\theta=0}^{T} \psi^{i,c}_a(\theta)\mathrm{d}\theta = \sum_{a\in\delta^{\text{in}}(t_i)} \int_{\theta=\tau_a}^{T} \psi^{i,c}_a(\theta-\tau_a)\mathrm{d}\theta + \sum_{v\in I:\ \mathfrak{b}_v \ge 0} \int_{\theta=0}^{T} \psi^{i,c}_v(\theta)\mathrm{d}\theta, \quad (18)$$

such that

$$\sum_{a\in\delta^{\text{in}}(v)} \int_{\beta=\tau_a}^{\theta} \psi^{i,c}_a(\beta-\tau_a)\mathrm{d}\beta - \sum_{a\in\delta^{\text{out}}(v)} \int_{\beta=0}^{\theta} \psi^{i,c}_a(\beta)\mathrm{d}\beta \ge 0, \quad \forall v \in I, i \in K, \theta \in \mathcal{T}, \quad (19)$$

$$0 \le \psi^c_a(\theta) = \sum_{i\in K} \psi^{i,c}_a(\theta) \le \mathfrak{u}_a, \quad \forall a \in A, \theta \in \mathcal{T}, \quad (20)$$

$$0 \le \psi^c_v(\theta) = \sum_{i\in K} \psi^{i,c}_v(\theta) \le \mathfrak{b}_v, \quad \forall v \in I, \theta \in \mathcal{T}, \quad (21)$$

$$\sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a \leq \mathfrak{b}_v \leq T \sum_{a\in\delta^{\text{in}}(v)} \mathfrak{u}_a, \quad \forall v \in I. \tag{22}$$

Here, equation (18) represents an objective function, which is to maximize the total amount of flow out from the source in continuous-time settings, which is sent to the sink and the intermediate nodes. The nonconservation of the flow is represented by equation (19). Equations (20)–(22) have their usual meanings as in Section 2.

Dynamic flow problems defined in Section 4 and Section 5 can be solved in continuous-time settings by using this natural transformation with their respective algorithms within the same time complexity.

7. Conclusion

If the amount of flow out from the source node is more than the minimum cut capacity, then the excess flow cannot reach the sink. To deal with this problem, maximum static, maximum dynamic, and maximum dynamic contraflow problems with the storage of excess flow at intermediate nodes have been studied in two-terminal general network.

In this paper, we have investigated the multicommodity flow models with intermediate storage in static as well as dynamic networks. We have introduced the maximum static and maximum dynamic multicommodity flow problems. We have presented a polynomial time algorithm to solve the maximum static multicommodity flow problem and a pseudopolynomial time algorithm for the maximum dynamic multicommodity flow problem by allowing the storage of excess flow at intermediate nodes. Moreover, we have presented an algorithm to solve the maximum dynamic multicommodity contraflow problem with symmetric as well as asymmetric transit times. By using natural transformation in multicommodity network, we solved the maximum dynamic flow and maximum dynamic contraflow problems with intermediate storage in continuous-time settings. To the best of our knowledge, the maximum multicommodity flow problems with intermediate storage and their solution strategies for the static flow, dynamic flow, and contraflow problems are introduced for the first time.

The universally maximum multicommodity flow problem (the earliest arrival flow problem) with intermediate storage is harder problem, which is of interest for the further research. This problem is $\mathcal{NP}$-hard even in case of two-terminal series parallel networks. Together with this, we are interested to work on maximum flow and earliest arrival flow problems with orientation-dependent transit times and flow-dependent transit times. We are also interested in implementing these techniques as a case study in Kathmandu road network.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

References

[1] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[2] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 1962.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithm and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[4] A. Ali, R. Helgason, J. Kennington, and H. Lall, "Technical note-computational comparison among three multicommodity network flow algorithms," *Operations Research*, vol. 28, no. 4, pp. 995–1000, 1980.

[5] A. A. Assad, "Multicommodity network flows-a survey," *Networks*, vol. 8, no. 1, pp. 37–91, 1978.

[6] J. L. Kennington, "A survey of linear cost multicommodity network flows," *Operations Research*, vol. 26, no. 2, pp. 209–236, 1978.

[7] A. Hall, S. Hippler, and M. Skutella, "Multicommodity flows over time: efficient algorithms and complexity," *Theoretical Computer Science*, vol. 379, no. 3, pp. 387–404, 2007.

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

[9] D. Khanal, U. Pyakurel, U. Pyakurel, and T. Dhamala, "Prioritized multi-commodity flow model and algorithm," in *Proceedings of the International Symposium on Analytic Hierarchy Process 2020 (ISAHP)*, London, UK, August 2020.

[10] U. Pyakurel and S. Dempe, "Network flow with intermediate storage: models and algorithms," *SN Operations Research Forum*, vol. 1, no. 4, 2020.

[11] U. Pyakurel, S. Wagle, and M. C. Adhikari, "Efficient lane reversals for prioritized maximum flow"" *International Journal of Innovative Science, Engineering & Technology*, vol. 7, pp. 354–363, 2020.

[12] U. Pyakurel and S. Dempe, "Universal maximum flow with intermediate storage for evacuation planning"," in *Dynamics of Disasters*, I. S. Kotsireas, A. Nagurney, P. M. Pardalos, and A. Tsokas, Eds., Springer, Berlin, Germany, 2021.

[13] 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, no. 3, pp. 101–133, 2018.

[14] 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.

[15] 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.

[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] 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.

[18] T. N. Dhamala, S. P. Gupta, D. P. Khanal, and U. Pyakurel, "Quickest multi-commodity flow over time with partial lane reversals," *Journal of Mathematics and Statistics*, vol. 16, no. 1, pp. 198–211, 2020.

[19] S. P. Gupta, D. P. Khanal, U. Pyakurel, and T. N. Dhamala, "Approximate algorithms for continuous-time quickest multi-commodity contraflow problem," *The Nepali Mathematical Sciences Report*, vol. 37, no. 1-2, pp. 30–46, 2020.

[20] U. Pyakurel, S. P. Gupta, D. P. Khanal, and T. N. Dhamala, "Efficient algorithms on multicommodity flow over time problems with partial lane reversals," *International Journal of Mathematics and Mathematical Sciences*, vol. 2020, Article ID 2676378, 13 pages, 2020.

[21] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960, http://www.jstor.org/stable/167547.

[22] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, John Wiley & Sons, Hoboken, NY, USA, 4th edition, 2010.

[23] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[24] H. N. Nath, U. Pyakurel, and T. N. Dhamala, "Network reconfiguration with orientation-dependent transit times," *International Journal of Mathematics and Mathematical Sciences*, vol. 2021, Article ID 6613622, 11 pages, 2021.