Available online at https://ganitjournal.bdmathsociety.org/

GANIT

GANIT: Journal of Bangladesh Mathematical Society

GANIT *J. Bangladesh Math. Soc.* 45.2 (2025) 001–017

DOI: https://doi.org/10.3329/ganit.v45i2.86700

Quickest Multi-commodity Flow Problem with Capacity Sharing

Durga Prasad Khanal[a], Shiva Prakash Gupta[b], Urmila Pyakurel[c], and Tanka Nath Dhamala*[c]

[a] *Saraswati Multiple Campus, Tribhuvan University, Kathmandu, Nepal*

[b] *Tri-Chandra Multiple Campus, Tribhuvan University, Kathmandu, Nepal*

[c] *Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal*

ABSTRACT

The quickest multi-commodity flow problem arises when more than one commodity is to be transported from the specific source nodes to corresponding sink nodes through the arcs in an underlying dynamic network within the minimum possible time. Sharing of the capacity of the bundle (common) arcs is one of the major issues for the multi-commodity flow problem. In this paper, we deal with the quickest multi-commodity flow problem by sharing the capacity of bundle arcs using proportional and flow-dependent capacity sharing techniques, which reduce the multi-commodity flow problem into single commodity flow problems. We present the polynomial and pseudo-polynomial algorithms to solve the problem by proportional and flow-dependent sharing, respectively. A three dimensional time-expanded layer graph is introduced to solve the problem with flow-dependent capacity sharing technique.

Received: February 8, 2024 **Accepted:** September 7, 2025 **Published Online:** December 30, 2025

Keywords: Multi-commodity; quickest flow; layer graph; proportional sharing; cost-scaling.

1 Introduction

Literature Review. The network flow problem is modelled by using a topological structure corresponding to some region in which entities are transmitted from one point to another point through some feasible routes. The transportation network is a key example of network topology, where road segments are depicted as arcs and their intersections as nodes. Vehicles or pedestrians on the roads are viewed as flows, with their starting points and destinations represented as the source and sink nodes, respectively.

In a dynamic network, arcs have capacities that limit the flow and transit times indicating the time needed to move the flow between nodes. The network flow over time problem, or dynamic flow,

*Corresponding author. *E-mail address:* tanka.nath.dhamala@gmail.com

is a significant topic of research in mathematical programming and operations research. This involves transferring flow from the supply area (source) to the demand area (sink), taking into account the required travel time. This issue is commonly applied to communication networks, highway and railway systems, supply and demand chains, and message routing problems. For the single commodity case, Ford and Fulkerson [1, 2] introduced the concept of flow over time problem and presented algorithm to solve the maximum flow over time problem. In this problem, the goal is to transfer the maximum possible flow from the origin/source to the destination/sink within a specified time frame. Khanal et al. [3] and Pyakurel et al. [4] solved the maximum dynamic flow problem with intermediate storage in a general network and abstract network structures, respectively, where excess flow that can not reach to the destination is stored at appropriate intermediate elements. Similarly, Dhamala et al. [5] solved the maximum flow problem with intermediate storage in lossy network topology. The solution strategy of the maximum dynamic flow model with speed and transit time variation incorporating the intermediate storage can be found in [6].

The quickest flow problem is an inverse problem of the maximum dynamic flow problem, which aims to transport a specified amount of flow from the origin/source to the destination/sink in the smallest possible time. Chen and Chin [7] and Rosen et al. [8] used the quickest path to transit the given amount of flow from the source to the sink in shortest possible time. They used the single quickest path in their model. Burkard et al. [9] presented a polynomial time algorithm of quickest flow problem by using binary search and Newton's methods. Fleischer and Tardos [10] provided an equivalent solution for the continuous time settings. Lin and Jaillet [11] solved the quickest flow problem applying the cost-scaling algorithm of Goldberg and Tarjan [12] within the same time complexity.

The multi-commodity network flow problem involves transmitting multiple commodities from their respective sources to corresponding sinks while ensuring optimal flow allocation without exceeding the arcs' capacity constraints. This problem was first introduced by Ford and Fulkerson [1], and many researchers have since extended the models and algorithms to incorporate various aspects such as maximum flow, maximum concurrent flow, quickest flow, and minimum cost flow. Multi-commodity flow problems differ significantly from single-commodity ones in bundle arcs, as they carry more than one commodity. Contrary to the multi-commodity cases, single-commodity models cancel out flows to avoid cycles in opposite directions. The well-known max-flow min-cut theorem, often used to calculate maximum flow for single-commodity problems, does not apply to multi-commodity flow problems. Additionally, if the capacities and flows on each arc are integers, single-commodity flow problems result in integer solutions. However, this does not necessarily hold true for multi-commodity flow problems. For further illustrations and applications, we refer to the book of [13], articles [14, 15, 16, 17, 18, 19] and references therein. Recently, Fan et al. [20] studied the minimum-cost multi-commodity flow problem on evolving networks where the network topology dynamically changes over time. Similarly, Lienkamp and Schiffer [21] solved a standard integer minimum-cost multi-commodity flow problem to obtain the passenger flow system optimum for an intermodal transportation system.

Given supplies at sources and demands at sinks, the quickest multi-commodity flow (QMCF) problem involves distributing various commodities from their respective sources to corresponding sinks through a designated network. The goal is to meet the total demand for each commodity within the shortest possible makespan. Hall et al. [22] proved that the multi-commodity flow over time problem is $\mathcal{NP}$-hard even in the case of series-parallel network or in the case of two commodities. Together with this, the $\mathcal{NP}$-hardness of the quickest multi-commodity flow problem with or without intermediate storage and simple paths is also found in it. Fleischer and Skutella [23] presented the length bounded approximation and condensed time-expanded graph to solve the QMCF problem in polynomial time complexity. For most of the real world problems, transit times are not fixed but flow-dependent. Köhler and Skutella [24] introduced the concept of load-dependent transit times by considering the total amount of flow on the arc as the load. Hall et al. [25] presented an FPTAS (fully polynomial time approximation scheme) for QMCF with inflow-dependent transit times, where transit time on arc depends on the inflow rate of the flow. Priority based multi-commodity flow problems and polynomial time solution strategies can be found in [26, 27]. Khanal et al. [28] introduced the proportional and flow dependent capacity sharing techniques to share the capacity of bundle arc for each individual commodity and used it to solve the maximum multi-commodity flow problem.

Research Gap. Solving quickest multi-commodity flow problem using two techniques of sharing capacity on bundle arcs - one based on incoming arc capacities (i.e., proportional capacity sharing) and another based on inflow rates of the flow (i.e., flow-dependent capacity sharing) remain unexplored in existing literature. Proportional capacity sharing simplifies the multi-commodity flow problem by breaking it down into independent single-commodity flow problems. This can be useful on solving the quickest multi-commodity flow problem using cost-scaling method which is not yet explored in the literature. Similarly, the application of a three-dimensional layer graph to solve the quickest multi-commodity flow problem remains unexamined.

Contributions. In this paper, we employ two capacity sharing techniques to solve the quickest multi-commodity flow (QMCF) problem, termed as proportional and flow-dependent capacity sharing. We introduce models for the QMCF problem using these techniques. To tackle the issue of proportional capacity sharing on arcs, we propose polynomial time algorithms using the cost-scaling method. Additionally, a pseudo-polynomial time algorithm is presented for solving the QMCF problem adopting flow-dependent capacity sharing on arcs. By introducing a three-dimensional layer graph, we provide solutions using flow-dependent capacity sharing technique. As far as we know, QMCF problem has been solved using these sharing techniques for the first time.

Organization of the Paper. The organization of the paper is as follows. By setting the basic notations, mathematical formulations of flow models and the three dimensional time-expanded layer graph are presented in Section 2. In Section 3, we present the QMCF problem with proportional capacity sharing. Our main result in this section is a polynomial time algorithm to solve the problem by using a cost-scaling approach. We introduce flow-dependent capacity sharing in Section 4 and present a pseudo-polynomial time algorithm to solve the QMCF problem. The paper is concluded in Section 5.

2 Mathematical Formulations

In this section, we set necessary notations and give mathematical formulations of different classes of multi-commodity flow problems. As most of the real world problems are associated with the shipment of multiple commodities, these models can be applied to solve transportation problems, demand supply chains and many more.

Basic Terminologies. Let us consider a set N of nodes in a graph with $|N| = n$ and another set $E \subseteq N \times N$ of links joining two nodes, known as arcs, with $|E| = m$. We represent $S \subset N$ and $D \subset N$ as the set of sources and sinks, respectively. For each commodity $i \in K = \{1, 2, \dots, k\}$, d_i denotes the demand which is routed through a unique source-sink pair $s_i - t_i$, where s_i belongs to the set S and t_i contained in the set D. Each arc $e = (v, w) \in E$ is associated with a capacity function $\mathfrak{b} : E \to \mathcal{R}^+$ which represents the upper limit of the flow on it, where $head(e) = w$ and $tail(e) = v$. Similarly, $\tau : E \to \mathcal{R}^+$ indicates the transit time function which refers the time needed to transfer the flow from node v to node w. The symbols $\delta^{out}(v)$ represents the set of outgoing arcs from node v whereas $\delta^{in}(v)$ represents the set of incoming arcs to node v. We represent $\mathcal{T} = \{0, 1, ..., T\}$ as a set of timeslots with time horizon T in discrete time settings. With these components, we represent the dynamic network as $\Omega = (N, E, K, \mathfrak{b}, \tau, d_i, S, D, T)$. For the static network, the time parameters T and τ are absent. The list of symbols are presented in Table 2.1.

Table 2.1: Symbols used throughout the paper.

Symbol	Meaning	Symbol	Meaning
Ω	Network topology	N	Set of nodes
E	Set of arcs	$\mathfrak{b}_e$	Capacity of an arc $e \in E$
τ_e	Transit time of an arc $e \in E$	K	Set of commodities
S	Set of source nodes s_i for $i \in K$	D	Set of destination nodes t_i for $i \in K$
T	Time horizon	d_i	Demand of Commodity $i \in K$
$\mathcal{T}$	Set of discrete time steps $\{0, 1, \dots, T\}$	$\delta^{out}(v)$	Sets of outgoing arcs from node v
$\delta^{out}(v)$	Sets of outgoing arcs from node v	$head(e)$	Head node of arc e
$tail(e)$	Tail node of arc e	φ_e^i	Static flow of commodity i on arc e
$f_e^i(\theta)$	Dynamic flow of commodity i on arc e at time $\theta \in \mathcal{T}$	$\alpha^i(e)$	$s_i - v$ path from source s_i to the tail node v of the bundle arc e
Ω^T	Time-expanded layer graph	c_e	Cost associated with arc e
I	Set of intermediate nodes $N \setminus \{S, D\}$	C	$\max_{e \in E}\{\tau_e\}$
$\gamma(e)$	Set of incoming arcs to bundle arc e	$\mathbf{P}_i(\theta)$	Set of paths from s_i at time θ

Static Flow Model. For a static network $\Omega = (N, E, K, \mathfrak{b}, d_i, S, D)$, the multi-commodity flow function φ is a set of non-negative flows $\varphi^i : E \to \mathcal{R}^+$ for each i with demand d_i satisfying the conditions (2.1 - 2.3).

$$\sum_{e \in \delta^{out}(v)} \varphi_e^i - \sum_{e \in \delta^{in}(v)} \varphi_e^i = \begin{cases} d_i & \text{if } v = s_i \\ -d_i & \text{if } v = t_i \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in N, i \in K \tag{2.1}$$

$$\sum_{i \in K} \varphi_e^i \leq \mathfrak{b}_e \ \forall e \in E \tag{2.2}$$

$$\varphi_e^i \geq 0 \ \forall e \in E, i \in K \tag{2.3}$$

Here, the notation φ_e^i represents the flow of commodity i on arc e. The boundedness of the arc flows by their capacities are represented in (2.2), known as bundle constraints. The non-negativity of the flows are presented in constraint (2.3). Similarly, the first two conditions in (2.1) represent supply/demand at source/sink nodes whereas the flow conservation at intermediate nodes is reflected by its third condition. Moreover, the cost of static flow φ can be defined as follows: if c_e^i be the cost coefficient associated with arc e and commodity i,

$$c(\varphi) = \sum_{e \in E} \sum_{i \in K} c_e^i \varphi_e^i \tag{2.4}$$

Dynamic Flow Model. Considering a dynamic network Ω with constant transit times τ on arc e, the dynamic multi-commodity flow function f is the collection of flows $f^i : E \times \mathcal{T} \to \mathcal{R}^+$, satisfying the constraints (2.5 - 2.8).

$$d_v^i = \sum_{e \in \delta^{out}(v)} \sum_{\theta=0}^{T} f_e^i(\theta) - \sum_{e \in \delta^{in}(v)} \sum_{\theta=0}^{T} f_e^i(\theta), \qquad \forall v \in N, i \in K \tag{2.5}$$

$$\sum_{e \in \delta^{out}(v)} \sum_{\theta=0}^{\beta} f_e^i(\theta) - \sum_{e \in \delta^{in}(v)} \sum_{\theta=0}^{\beta} f_e^i(\theta) \leq 0, \qquad \forall v \notin \{s_i, t_i\},\ i \in K,\ \beta \in \mathcal{T} \tag{2.6}$$

$$\sum_{i \in K} \sum_{\theta=0}^{T} f_e^i(\theta) \leq \mathfrak{b}_e, \qquad \forall e \in E \tag{2.7}$$

$$f_e^i(\theta) \geq 0, \qquad \forall i \in K \quad \text{and} \quad e \in E. \tag{2.8}$$

where,

$$d_v^i \;=\; \begin{cases} d_i & \text{if } v = s_i \\ -d_i & \text{if } v = t_i \qquad \forall i \in K \\ 0 & \text{otherwise} \end{cases}$$

Here, $f_e^i(\theta)$ represents the flow rate of commodity i on arc e at time $\theta \in \mathcal{T}$. The constraints outlined in (2.5) ensure the supply and demand at the sources and sinks, as well as the conservation of flow at intermediate nodes during the time horizon T. Constraints in (2.6) account for the lack of flow conservation at intermediate nodes at any time step $\beta \in \mathcal{T} = \{0, 1, ..., T\}$. Meanwhile, the constraints in (2.7) are restricted by capacities, and (2.8) ensures non-negativity. The cost of a discrete dynamic flow is defined by

$$c(f) = \sum_{e \in E} \sum_{i \in K} c_e^i \sum_{\theta=0}^{T} f_e^i(\theta). \tag{2.9}$$

Time-expanded Layer Graph. For multi-commodity flow, the time-expanded layer graph can be constructed as a three-dimensional graph that replicates nodes from a static network for each discrete time step and commodity. As in [29], consider a dynamic network Ω with integral transit times on arcs and time horizon T. For the purpose of T-time-expanded layer graph Ω^T, we creating $T+1$ copies of the node set N labeled $N(0), N(1), ..., N(T)$. In this setup, the θ^{th} copy of node v is denoted as $v(\theta)$, where $\theta \in \mathcal{T}$. For each time step $\theta \in \{0, 1, ..., T - \tau_e\}$, we create the copy of an arc $e = (v, w) \in E$ from $v_i(\theta)$ to $w_i(\theta + \tau_e)$ and represent it as $e_i(\theta)$, which has the same capacity as arc e if the flow on arc is of single commodity and shared capacity if arc e is a bundle arc. Moreover, arc from $v_i(\theta)$ to $v_i(\theta + 1)$ with infinite capacity is known as holdover arc, which holds the flow for a unit time interval $[\theta, \theta + 1)$. In continuous time settings, $T + 1$ copies of N are labeled in layer graph Ω^T as $N[0, 1), N[1, 2), ..., N[T, T + 1)$ where θ^{th} copy of node v is labeled as $v[\theta, \theta + 1)$ for $\theta \in \mathcal{T}$. For graphical representation, the three-dimensional layer graph Ω^T is displayed with node N, time T, and commodity K as the coordinate axes in Figure 2.1. Each commodity $i \in K$ forms a horizontal layer of graphs in vertical axis. Figure 2.1(a) shows a two-commodity network, where commodity-1 is transported from s_1 to t_1 and commodity-2 from s_2 to t_2. Arc (x, y) is a bundle arc carrying both commodities. Figure 2.1(b) represents the time-expanded layer graph of Figure 2.1(a).

Quickest Flow Problem. For the given network, quickest multi-commodity flow problem intends to find a minimum possible time to satisfy all demands d_i from s_i to t_i. Mathematically, it can be presented as

$$\begin{cases} T_i^* = \min \frac{d_i + \sum_{e \in E} \tau_e \varphi_e^i}{|\varphi^i|} - 1, \qquad (|\varphi^i| \neq 0) \\ \text{satisfying the constraints } \; (2.1 - 2.3) \end{cases} \tag{2.10}$$

where, φ_e^i denotes the feasible static flow of i^{th}-commodity on arc e with flow value $|\varphi^i|$. Similarly, τ_e is the arc cost (or transit time) of arc e. The quickest time to satisfy all the demands d_i is $T^* = \max\{T_i^* \mid i \in K\}$.

3 QMCF with Proportional Capacity Sharing

The commodity-wise sharing of a bundle arc capacity creates a significant challenge in multi-commodity flow problems. The resource directive decomposition method treats this problem as a

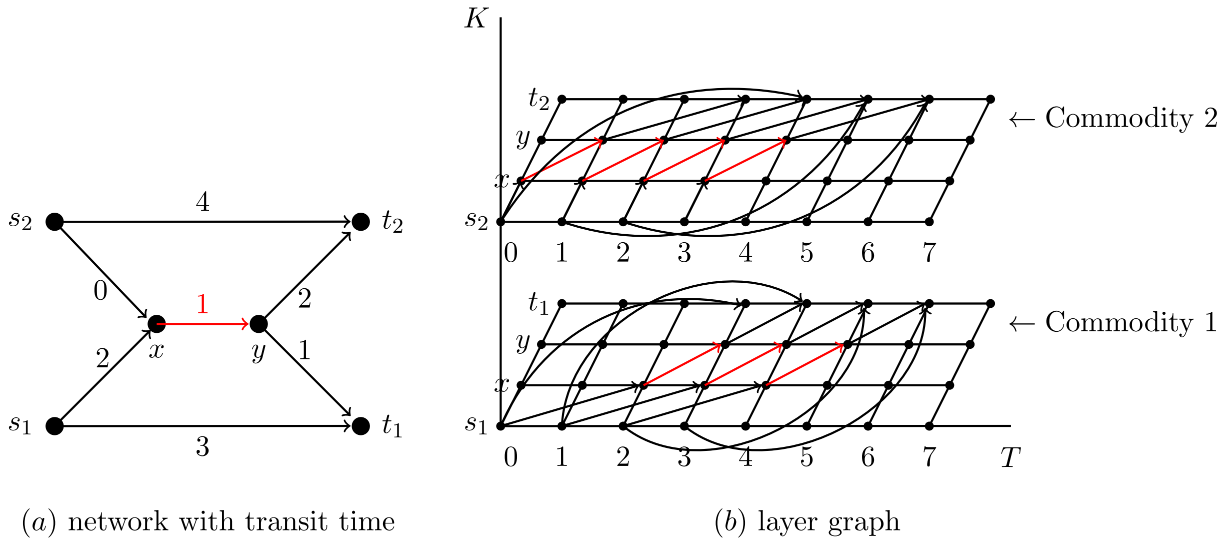

Figure 2.1: Time-expanded layer graph Ω^T in (b) for a two-commodity network (a).

capacity allocation problem. Initially, it distributes the available capacity among various commodities, using insights gained from previous solutions. The method then reallocates these capacities to enhance overall system efficiency and reduce costs. During each iteration, it resolves k single-commodity flow problems. The total capacity assigned across all commodities always equals the original arc capacity, ensuring resource constraints are respected.

As in Khanal et al. [28], we reallocate the bundle arc capacity by a proportional capacity sharing approach, where the capacity is shared proportionally to each commodity and reduces the multi-commodity flow problem to k independent single commodity flow problems (c.f. [29]). The total capacity assigned across all commodities always equals the original arc capacity and the capacity of each individual commodity restricts the flow of the commodity on it. Hereafter, we present the proportional capacity sharing technique and formulate the QMCF problem. With the help of cost-scaling approach of [11], we present a polynomial time algorithm to solve the problem.

3.1 Proportional Capacity Sharing

In single-commodity flow problems, all flow is treated as identical. However, in multi-commodity flow problems, flows are differentiated based on their unique attributes, meaning the demand of one commodity cannot fulfill that of another. This distinction increases the complexity of the problem. Consequently, multi-commodity flow problems are more challenging than their single-commodity counterparts due to the constraints imposed by shared capacities on bundle arcs. When multiple commodities traverse the same bundle arc, the issue of capacity allocation arises.

To address this, we assume that every commodity is associated with a unique source-sink pair. The intermediate nodes, defined as $I = N \setminus \{S, D\}$, exclude both the source set S and the destination set D, ensuring I, S, and D are mutually exclusive. Arcs connecting pairs of intermediate nodes are considered bundle arcs if they transport more than one commodity. For instance, a bundle arc $e = (v, w)$ satisfies $v, w \in I$, though the reverse is not necessarily true. To allocate the capacity of a bundle arc effectively, we introduce a proportional sharing method. This technique relies on the bottleneck capacities of the paths corresponding to each commodity i, which originate from their respective sources s_i and lead to the bundle arc e. For a bundle arc $e = (v, w)$ with capacity $\mathfrak{b}_e$, the proportional sharing of capacity $\mathfrak{b}_e$ to each commodity $i \in K$ is,

$$\mathfrak{b}_e^i = \frac{\mathfrak{b}_a^i}{\sum\limits_{a \in \alpha^i(e): i \in K} \mathfrak{b}_a^i} \mathfrak{b}_e \tag{3.1}$$

where, $\alpha^i(e)$ is the $s_i - v$ path from source s_i to the tail node v of the bundle arc e and $a \in \alpha^i(e)$ is an arc with bottleneck capacity of path $\alpha^i(e)$. Here, $\mathfrak{b}_e^i$ represents the portion of the capacity of arc e allocated for the commodity i. Clearly, $\sum\limits_{i \in K} \mathfrak{b}_e^i = \mathfrak{b}_e$. In Figure 2.1(b), red color arcs of the layer graph share the capacity of the bundle arc (x, y) in Figure 2.1(a) for both commodities.

When distributing the capacity of a bundle arc, fractional values may arise, expressed as $\mathfrak{b}_e^i = \text{int}(\mathfrak{b}_e^i) + \text{fra}(\mathfrak{b}_e^i)$, where the capacity is split into an integral and a fractional component. Unless $\mathfrak{b}_e^i < 1$ with no alternative path available for commodity i, fractional capacities can be converted to integral capacities using the following method ([29]):

- Calculate the total sum of fractional components, $\sum_i \text{fra}(\mathfrak{b}_e^i)$. If this sum equals p, round up the p largest fractional components using ceiling function $\lceil . \rceil$. The remaining fractional components are rounded down using the floor function $\lfloor . \rfloor$.
- In cases where multiple commodities share the same fractional value, preference is given to the commodity with the largest integral capacity.
- If the integral capacities are identical as well, priority shifts to the commodity with the highest demand. Should the demand also be equal, any of the tied commodities may be rounded up arbitrarily.

For example, let a bundle arc e has capacity $\mathfrak{b}_e = 21$ and sharing capacities $\mathfrak{b}_e^i$ for five commodities are 3, 2.8, 3.5, 4.5, and 7.2. Then, being $\sum \text{fra}(\mathfrak{b}_e^i) = 2$, two sharing capacities out of five are rounded up. Having the greatest fractional part, 2.8 is first rounded up by 3. For the second one, fractional part 0.5 occurs in two commodities. Priority is given to the greatest integral part among these two, and so 4.5 is rounded up to 5. The rest are rounded below. Thus capacities assigned to respective commodities are $\mathfrak{b}_e^i = 3, 3, 3$, 5 and 7. It is to be careful that if shared capacity of some commodity i in bundle arc is less than one and has no alternative path, floor function can block the flow. If so, we can accept the fractional capacity or can use flow-dependent capacity sharing described in Section 4.

3.2 Problem Formulation

Ford and Fulkerson [2] presented a generic method to solve the maximum flow over time problem for a single source single sink network, known as temporally repeated flow formulation. The quickest flow problem is inherently linked to the maximum flow problem, where the maximum flow is repeated over the time from origin to destination until the demand is fully met. By transforming this formulation, Lin and Jaillet [11] presented the quickest flow problem for the single commodity flow with fractional programming. We formulate the quickest multi-commodity flow problem by extending the concept of single commodity flow problem of [11] and incorporating the proportional capacity sharing as follows.

$$\begin{cases} T_i^* = \min \dfrac{d_i + \sum\limits_{e \in E} \tau_e \varphi_e^i}{|\varphi^i|} - 1, \qquad (|\varphi^i| \neq 0) \\ \text{subject to, } \; \mathfrak{b}_e^i = \dfrac{\mathfrak{b}_a^i}{\sum\limits_{a \in \alpha^i(e): i \in K} \mathfrak{b}_a^i} \mathfrak{b}_e \; \text{ and the constraints } (2.1 - 2.3). \end{cases} \tag{3.2}$$

The new constraint $\mathfrak{b}_e^i$ obtained from (3.1) reduces the multi-commodity flow problem into k independent single commodity sub-problems so that the QMCF problem (3.2) with proportional capacity sharing can be solved by solving k independent single commodity flow problems. The objective of the problem is to find the minimum time T_i^* satisfying the given demand d_i for each commodity $i \in K$. By setting $T^* = \max\{T_i^* \mid i \in K\}$, it provides the quickest time satisfying all the demands.

3.3 Solution Strategy: The Cost-scaling Approach

In this subsection, our aim is to present the cost-scaling algorithm for the QMCF problem presented in (3.2) with proportional capacity sharing on the bundle arcs depending on the algorithms of [11]. The algorithmic framework is presented in Algorithm 1 herein.

Algorithm 1: The cost-scaling algorithm for QMCF

Input: Multi-commodity dynamic network $\Omega = (N, E, K, \mathfrak{b}, \tau, d_i, S, D, T)$.

Output: Quickest time to satisfy all demands d_i with proportional capacity sharing on Ω.

1. Create k independent sub-problems by sharing the capacity on bundle arcs using Equation (3.1).
2. Initialize the node potentials $\pi(v) = 0$ for all nodes $v \in N$, set the initial flow $\varphi_e^i = 0$ for all edges $e \in E$, and define ϵ as $\epsilon = C = \max_{e \in E}\{\tau_e\}$.
3. Transform the 2ϵ-optimal flow into an ϵ-optimal flow.
4. Minimize the gap between T_i and the difference in potential $\pi(s_i) - \pi(t_i)$.
5. Halve ϵ and repeat Steps 3 and 4 if $\epsilon \geq \frac{1}{8n_i}$.
6. In the residual network $\bar{\Omega}$, if T_i exceeds the cost of the shortest simple path, then send the maximum static flow φ^i from s_i to t_i to saturate the flow within the subnetwork Ω' which includes those arcs lying on shortest path in residual network $\bar{\Omega}$.
7. Finally, determine T as the maximum of T_i for all $i \in K$.

To minimize the ratio

$$T_i = \frac{d_i + \sum_{e \in E} \tau_e \varphi_e^i}{|\varphi^i|} - 1,$$

dual variables related to the flow conservation constraints are introduced, along with node potentials π and the residual network $\bar{\Omega}$. For each arc $e = (v, w)$ in $\bar{\Omega}$, reduced cost $c(e, \pi)$ is calculated by $c(e, \pi) = \pi(w) - \pi(v) + \tau_e$ which we consider not less than $-\epsilon$, for some $\epsilon > 0$.

The algorithm begins with constructing k independent sub-problems for each commodity $i \in K$ by sharing the capacity of bundle arcs proportionally. Then we initialize each node potential as zero, arc flow as zero and $\epsilon = C = \max_{e \in E}\{\tau_e\}$, for all $e \in E$. By assigning

$$\varphi_e^i = \begin{cases} 0 & \text{if } c(e, \pi) > 0 \\ \mathfrak{b}_e^i & \text{if } c(e, \pi) < 0 \end{cases} \quad \forall e \in E,$$

each 2ϵ-optimal flow is modified to an ϵ-optimal flow. The Push/Relabel algorithm is applied to active nodes with positive excess flow. It is important to note that a static flow φ^i is considered ϵ-optimal if its reduced cost is no less than $-\epsilon$. To reduce the gap between T_i and $\pi(t_i) - \pi(s_i)$, extra flow is created at the source node s_i, and the admissible flow is directed through the arcs in $\bar{\Omega}$ to the sink node t_i. Node potentials are adjusted as needed. Then, ϵ is halved, and this process is repeated until ϵ drops below $\frac{1}{8n_i}$, where n_i denotes the number of nodes corresponding to commodity i. If the T_i obtained from the scaling phases surpasses the cost of the simple shortest path from s_i to t_i in $\bar{\Omega}$, the flow is then maximized by sending the highest feasible flow from s_i to t_i within the subnetwork Ω'. The subnetwork Ω' is formed from the residual network $\bar{\Omega}$, consisting of arcs that belong to some shortest path between s_i and t_i. Finally, T is defined as the maximum of T_i for all $i \in K$, representing the quickest possible time to fulfil all demands in the original network Ω.

Before presenting the correctness of Algorithm 1, we establish the optimality condition for the QMCF problem. For simplicity, we represent $|\varphi^i|$ as the parameter y^i and define $h(y^i) = \sum_{e \in E} \tau_e \varphi_e^i$. Using this notation, we can rewrite the objective function of the problem in (3.2) as

$$T_i^* = \min \frac{d_i + h(y^i)}{y^i} - 1 = \min T_i^*(y^i), \quad y^i > 0$$

Lemma 1. *The function $T_i^*(y^i)$ has local minimum at flow value y^i if and only if*

$$-(c_{t_i - s_i} + 1) \le T_i^*(y^i) \le c_{s_i - t_i}$$

where, $-c_{t_i-s_i}$ and $c_{s_i-t_i}$ represent the costs associated with the shortest paths from t_i to s_i and from s_i to t_i in the residual network of flow y^i, respectively.

Proof. The function $T_i^*(y^i)$ is local minimum at y^i if for arbitrary small $\epsilon > 0$,

$$T_i^*(y^i) = \frac{d_i + h(y^i)}{y^i} - 1 \le \frac{d_i + h(y^i + \epsilon)}{y^i + \epsilon} - 1 = T_i^*(y^i + \epsilon) \tag{3.3}$$

and,

$$T_i^*(y^i) = \frac{d_i + h(y^i)}{y^i} - 1 \le \frac{d_i + h(y^i - \epsilon)}{y^i - \epsilon} - 1 = T_i^*(y^i - \epsilon). \tag{3.4}$$

In this context, $h(y^i + \epsilon)$ denotes the increase in the flow value y^i by ϵ in the cost function $h(y^i)$. This indicates the sending of an additional ϵ amount of flow from s_i to t_i while maintaining the minimum-cost flow. To achieve this, the extra flow is routed through the shortest path in the residual network $\bar{\Omega}$ at a cost of $\epsilon\, c_{s_i-t_i}$. Consequently, $h(y^i + \epsilon) = h(y^i) + \epsilon\, c_{s_i-t_i}$. With this relation, Equation (3.3) becomes

$$\begin{aligned}
&\frac{d_i + h(y^i)}{y^i} - 1 \le \frac{d_i + h(y^i) + \epsilon\, c_{s_i-t_i}}{y^i + \epsilon} - 1 \\
&\Rightarrow c_{s_i-t_i} \ge \frac{d_i + h(y^i)}{y^i} \\
&\therefore\ T_i^*(y^i) = \frac{d_i + h(y^i)}{y^i} - 1 < \frac{d_i + h(y^i)}{y^i} \le c_{s_i-t_i}
\end{aligned} \tag{3.5}$$

In a similar manner, $h(y^i - \epsilon)$ denotes the reduction in the flow value y^i by ϵ in the cost function $h(y^i)$. This implies the return of an ϵ amount of flow from t_i to s_i, incurring a cost of $\epsilon\,(-c_{t_i-s_i})$. Thus, $h(y^i - \epsilon) = h(y^i) - \epsilon\,(-c_{t_i-s_i})$. Using this relation in Equation (3.4), we get

$$\begin{aligned}
&-c_{t_i-s_i} \le \frac{d_i + h(y^i)}{y^i} \\
&\Rightarrow -c_{t_i-s_i} - 1 \le \frac{d_i + h(y^i)}{y^i} - 1 \\
&\therefore\ T_i^*(y^i) = \frac{d_i + h(y^i)}{y^i} - 1 \ge -(c_{t_i-s_i} + 1)
\end{aligned} \tag{3.6}$$

□

Theorem 1. *The temporally repeated flow of a feasible static flow φ^i, for each commodity i, is considered optimal if and only if φ^i is a minimum-cost flow with a flow value of $|\varphi^i|$ and satisfies*

$$-(c_{t_i-s_i} + 1) \le \frac{d_i + \sum_{e \in E} \tau_e \varphi_e^i}{|\varphi^i|} - 1 \le c_{s_i-t_i}. \tag{3.7}$$

Proof. For the QMCF problem in (3.2), if we consider the flow value $|\varphi^i| = y^i$ as a parameter, then the objective function is defined as one less than the sum of $\frac{d_i}{y^i}$ and $\frac{1}{y^i}$ times the minimum-cost flow problem. As in [29], the minimum-cost flow problem can be expressed as follows.

$$\begin{cases} h(y^i) = \min \sum\limits_{e\in E} \tau_e \varphi_e^i \\ \text{subject to, } \ \mathfrak{b}_e^i = \dfrac{\mathfrak{b}_a^i}{\sum\limits_{a\in\alpha^i(e):i\in K} \mathfrak{b}_a^i}\mathfrak{b}_e \ \text{ and the constraints } (2.1-2.3) \end{cases} \tag{3.8}$$

For the problem in (3.2), objective function is to minimize

$$T_i^*(y^i) = \frac{d_i + h(y^i)}{y^i} - 1, \ y^i > 0.$$

According to Lemma 1, the flow value y^i represents a local minimum for the function $T_i^*(yi)$ if and only if the inequality $-(c_{t_i-s_i}+1) \leq T_i^*(yi) \leq c_{s_i-t_i}$ is satisfied. Additionally, the problem described in (3.8) is a linear programming problem aimed to minimizing costs, and $h(y^i)$ is a convex function that is piecewise linear in y^i. Consequently, $T_i^*(yi)$ exhibits a unimodal nature, and so a local minimum of $T_i^*(yi)$ is also the global minimum, [11].

Let $\bar{y}^i$ be the optimal flow value for $T_i^*(yi)$. In this case, the inequality $-(c_{t_i-s_i}+1) \leq T_i^*(\bar{y}i) \leq c_{s_i-t_i}$ must hold. Furthermore, if $\bar{\varphi}^i$ is a feasible flow with a value of $\bar{y}^i$, then $h(\bar{y}^i) = \min \sum\limits_{e\in E} \tau_e \bar{\varphi}_e^i$ if and only if $\bar{\varphi}^i$ represents a minimum cost flow. As a result,

$$T_i^*(\bar{y}^i) = \frac{d_i + \sum\limits_{e\in E} \tau_e \bar{\varphi}_e^i}{\bar{y}^i} - 1$$

if and only if $\bar{\varphi}^i$ is a minimum cost flow with flow value $\bar{y}^i$. Thus, the optimality of a feasible flow $\bar{\varphi}^i$ with a value of $\bar{y}^i$ is possible only if $\bar{\varphi}^i$ is the minimum cost flow and

$$-(c_{t_i-s_i}+1) \leq \frac{d_i + \sum\limits_{e\in E} \tau_e \bar{\varphi}_e^i}{\bar{y}^i} - 1 \leq c_{s_i-t_i}.$$

Therefore, the temporally repeated flow of the static flow $\bar{\varphi}^i$ is an optimal solution to the QMCF, considering proportional capacity sharing on the bundle arcs for all $i \in K$. □

Theorem 2. *The QMCF problem can be solved by Algorithm 1 using proportional capacity sharing on bundle arcs correctly.*

Proof. We prove the theorem in two parts. In the first part, we show the existence of minimum cost static flow φ^i and in the second part, we show that the flow is optimal quickest flow.

Given our assumption that all transit times on the arcs are integers, the arc transit times/costs must also be integers. Let π denote the node potential, and $c(e,\pi)$ represent the reduced cost in the residual network $\bar{\Omega}$. To allow for negative reduced costs, we select a value $\epsilon > 0$ such that $-\epsilon \leq c(e,\pi) < 0$. Any flow φ^i that meets this condition is termed ϵ-optimal flow.

To solve the problem, we begin with a large $\epsilon = C$ and halve it in each iteration to ensure the modified flow φ^i and node potential π remain ϵ-optimal. This process continues until ϵ becomes very small, ensuring that the sum of the reduced costs through cycles in $\bar{\Omega}$ is greater than -1. Since all arc costs are integers, there must be no negative cycles in $\bar{\Omega}$ with positive flow. Hence, the flow φ^i is the minimum cost flow.

To show that condition (3.7) is satisfied when the algorithm ends, an additional step is incorporated into each scaling phase so that $\pi(s_i) - \pi(t_i)$ accurately approximates both $-(c_{t_i-s_i}+1)$ and $c_{s_i-t_i}$. To reduce the gap between $\frac{d_i + \sum_{e\in E} \tau_e \varphi_e^i}{|\varphi^i|} - 1$ and $\pi(s_i) - \pi(t_i)$ by half in each iteration, extra flow is pushed from s_i to t_i, reducing the gap to less than 1 by the end of the scaling phase. Given that arc costs are integers, an optimal solution is achieved by solving an additional maximum flow problem. This process continues until the optimal solution for each commodity $i \in K$ is obtained. Consequently, Algorithm 1 correctly solves the QMCF problem with proportional capacity sharing.

□

Theorem 3. *The QMCF with proportional capacity sharing is polynomial time solvable by Algorithm 1.*

Proof. The computational time of Step 1 of Algorithm 1 is $O(m|K|)$, where $|K| = k$ signifying the number of commodities. The time complexity of Steps 3 and 4 is $O(n^3)$. Because of the cost-scaling phase in Step 5, Steps 3 and 4 are repeated up to $O(\log(nC))$ times, [11]. Similarly, the running time of Step 6 is $O(n^3 \log(n))$. Therefore, the overall running time of Algorithm 1 is $O(n^3 m|K|\log(nC))$, which is polynomial. Moreover, the solution obtained is an approximate optimal because of the use of ceiling and floor functions while sharing the capacity proportionally. □

Example 1. *Consider a two commodity network with demands $d_1 = 35$ for Commodity-1 and $d_2 = 25$ for Commodity-2 (c.f. Figure 3.1(a)). Flows of Commodity-1 and Commodity-2 routed via s_1-t_1 paths and s_2-t_2 paths, respectively. We aim to use proportional capacity sharing on the bundle arc (x, y) which provides the capacity of 2 units each for both commodities (as shown in Figure 3.1(b)). This problem is now converted to two independent sub-problems as follows.*

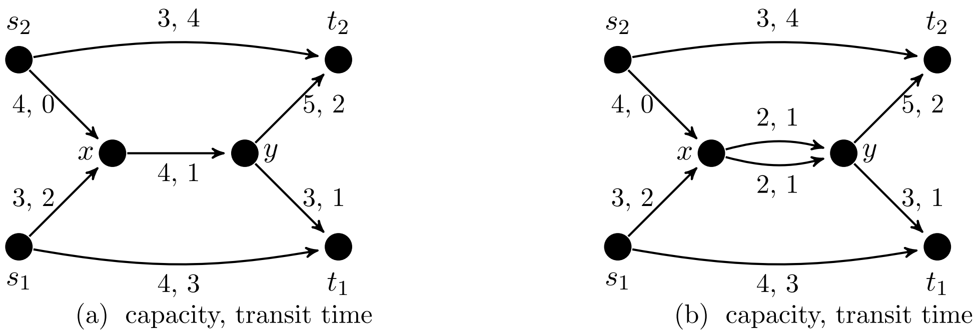

Figure 3.1: (b) represents proportional capacity sharing of (a).

Sub-problem 1: The quickest time to satisfy the demand-supply $d_1 = 35$ from s_1 to t_1 on the network 3.1(b) is,

$$T_1 = \min \frac{d_1 + \sum_{e\in A} \tau_e \varphi_e^i}{|\varphi^i|} - 1 = \frac{35 + 4 + 2 + 2 + 12}{6} - 1 = 8.16.$$

Sub-problem 2: As above, the quickest time satisfy the demand-supply of $d_2 = 25$ from s_2 to t_2 on the network 3.1(b) is,

$$T_2 = \min \frac{d_2 + \sum_{e\in A} \tau_e \varphi_e^i}{|\varphi^i|} - 1 = \frac{25 + 0 + 2 + 4 + 12}{5} - 1 = 7.6.$$

Therefore, the quickest time (in an integer) to satisfy both demands is $T = 9$.

4 QMCF with Flow-Dependent Capacity Sharing

In Section 3, we solved the QMCF problem by sharing the capacity on bundle arcs proportionally, where the shared capacity remains constant at each time step θ. In this section, our aim is to introduce the flow-dependent capacity sharing technique, where the commodity-wise shared capacity of a bundle arc is based on the inflow rate of the commodities in its incoming arcs. At any given time θ, the flow-dependent capacity sharing of a bundle arc $\mathfrak{b}_e$ to each commodity $i \in K$ is determined as follows.

$$\mathfrak{b}_e^i(\theta) = \frac{f_a^i(\theta - \tau_a)}{\sum\limits_{a \in \gamma(e): i \in K} f_a^i(\theta - \tau_a)} \mathfrak{b}_e. \tag{4.1}$$

Here, $\gamma(e)$ is a set of incoming arcs to bundle arc e, where $a \in \gamma(e)$ implies that $head(a) = tail(e)$, and $\mathfrak{b}e^i(\theta)$ denotes the share of capacity of arc e allocated to commodity i at time θ. The sum of the shared capacities $\mathfrak{b}e^i(\theta)$ across all commodities $i \in K$ equals the given capacity of arc e. Mathematically, this is expressed as $\sum\limits_{i \in K} \mathfrak{b}_e^i(\theta) = \mathfrak{b}_e$. In cases where the flow-dependent capacities are fractional, they can be converted into integer values as described in Subsection 3.1, even when $\mathfrak{b}e^i(\theta) < 1$ and there are no alternative paths for commodity i. As in [29], the QMCF problem with flow-dependent capacity sharing can be introduced as follows.

$$\begin{cases} \min T \\ \text{subject to the constraints (4.1) and (2.5 - 2.8)} \end{cases} \tag{4.2}$$

The constraint in (4.1) plays an important role in the QMCF problem (4.2). One of the reasons is that, a multi-commodity flow problem is reduced to k single commodity sub-problems at each time θ and can be solved as the single commodity flow problems. On the other hand, shared capacity $\mathfrak{b}_e^i(\theta)$ is not depending on the capacity $\mathfrak{b}_a$ of predecessor arc but on the flow entering on arc e through predecessor arcs a so that the absence of any commodity at any time step θ helps to increase the flow of other commodities. The quickest time to satisfy the given demand d_i is $T^* = \max\{T_i^* \mid i \in K\}$.

As in [29], the solution strategy starts by constructing temporal paths, denoted as $P(\theta) \in \mathbf{P}_i(\theta)$, where $P(\theta)$ represents a path that originates from s_i at time θ and reaches to t_i at time $\theta + \tau_P$ via arcs $e \in P$. The time-expanded layer graph is the way to visualize such paths. We define $\mathcal{P}(\theta)$ as $\{\cup \mathbf{P}_i(\theta) : i \in K\}$, which represents a set of all temporal paths starting at time θ. Similarly, $\tau'_P = \min\{\tau_P : P \in \mathbf{P}_i, \ \forall i\}$ is the length of the shortest path in the network. Let us represent $\varphi^{i,P}(\theta)$ as the time-dependent static flow on the temporal paths $P(\theta) \in \mathbf{P}_i(\theta)$. Due to flow-dependent capacity sharing on bundle arcs, amount of flow in such paths may vary over time. For any two temporal paths of different commodities $i, j \in K$, if paths $P_i(\theta_1)$ and $P'_j(\theta_2)$ meet at $u(\theta)$ with $\theta_1, \theta_2 \leq \theta$, then capacity along the arc $e = (u(\theta), v(\theta + \tau_e))$ is shared by employing flow-dependent capacity sharing.

Initially, we construct a time-expanded layer graph with the time horizon $T = \tau'_P$ and obtain the commodity-wise flow using flow-dependent capacity sharing. The time horizon is then increased by one unit (i.e., $T = \tau'_P + 1$) at each iteration as long as the demand of each commodity is met. The flow of commodities whose demand is already met is set to 0 in subsequent constructions of the time-expanded layer graph. Algorithm 2 presented hereafter is the stepwise procedure to find QMCF with flow-dependent capacity sharing.

Theorem 4. *Algorithm 2 provides an approximate solution to the quickest MCF problem with flow-dependent capacity sharing in pseudo-polynomial time.*

Proof. In Step 2(a), flow-dependent capacity sharing is used to distribute the bundle arc capacity, and the integer solution is obtained by applying the ceiling ($\lceil \cdot \rceil$) and floor ($\lfloor \cdot \rfloor$) functions. As a result, Algorithm 2 produces an approximate solution. The shared capacity in the bundle arc depends on

Algorithm 2: Flow-dependent QMCF algorithm

Input: Multi-commodity dynamic network $\Omega = (N, E, K, \mathfrak{b}, \tau, d_i, S, D, T)$.

Output: Quickest time with flow-dependent capacity sharing on Ω.

1. Initialize the time horizon: $T = \tau'_P = \min\{\tau_P : P \in \mathbf{P}_i, \forall i\}$ and set the flow for each commodity to zero: $f^i(T) = 0$.
2. **While** the flow for any commodity f^i is less than its demand d_i for all $i \in K$, **do**
   (a) Construct the time-expanded layer graph of time T and calculate the static flow φ^i by sharing the capacity on bundle arcs of temporal paths using Equation (4.1).
   (b) Calculate the flow value of each commodity within the current time horizon T as $f^i(T)$.
   (c) Set $\varphi^i = 0$ if $f^i(T) \geq d_i$.
   (d) Update $f^i(T) = f^i(T) + \varphi^i$ and $T = T + 1$.
3. $T =$ Quickest time to satisfy all the demands d_i.

the inflow rate from predecessor arcs, and the sharing process continues on the layer graph until all the demand d_i for each commodity $i \in K$ is fulfilled. Consequently, the algorithm's running time is influenced by the demand d_i, leading to a pseudo-polynomial time complexity for solving the quickest multi-commodity flow over time problem with flow-dependent capacity sharing.

□

Example 2. *Consider the network from Figure 3.1 of Example 1 having the same demands $d_1 = 35$ and $d_2 = 25$ for Commodity-1 and Commodity-2, respectively. Here, we solve the QMCF problem by using flow-dependent capacity sharing on bundle arc. Capacity on bundle arc (x, y) is shared after 2 unit times because of the absence of commodity-1. Similarly, after $T = 5$ only Commodity-1 is transported from the bundle arc due to the absence of Commodity-2, (see Figure 4.1). At each time θ, the capacities after flow-dependent capacity sharing on the arc (x, y) are presented in Figure 4.1. The quickest time to satisfy both demands is $T = 8$. This can be presented by using a time-expanded layer graph as follows.*

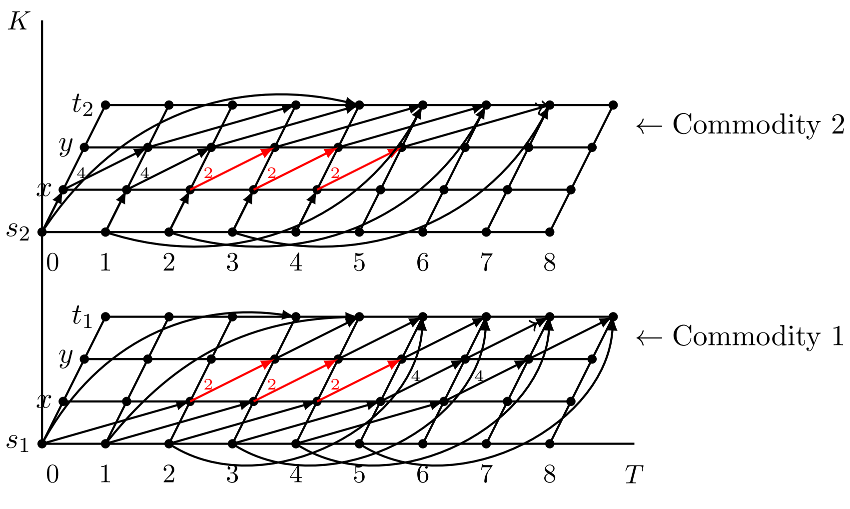

Figure 4.1: The time-expanded layer graph with flow-dependent capacity sharing.

Table 4.1 represents the flow pattern of two commodities that reach the destination with different time horizons T. Due to the absence of Commodity-1 in the first two time steps, only Commodity-2 with flow value 4 is transported from path $s_2 - x - y - t_2$. Similarly, only Commodity-1 is transported in path $s_1 - x - y - t_1$ with flow value 3 after the demand of Commodity-2 is fulfilled.

Table 4.1: Transshipment of flow in paths.

Time Horizon	Commodity-1 ($d_1 = 35$)				Commodity-2 ($d_2 = 25$)			
	s_1-t_1	s_1-x-y-t_1	flow	cumf.	s_2-x-y-t_2	s_2-t_2	flow	cumf.
3	4	-	4	4	4	-	4	4
4	4	2	6	10	4	3	7	11
5	4	2	6	16	2	3	5	16
6	4	2	6	22	2	3	5	21
7	4	3	7	29	2	3	5	26
8	4	3	7	36	-	-	-	-

cumf. = cumulative flow

Although the time complexity in the worst case analysis for the QMCF problem with flow-dependent capacity sharing is weaker (pseudo-polynomial) than the proportional capacity sharing (polynomial), the quickest time to satisfy the given demand by flow-dependent capacity sharing is better than the proportional capacity sharing. In Figure 4.2, we compare the change in cumulative flow with respect to time for Commodity-2 by using proportional as well as flow-dependent capacity sharing graphically. Comparison for Commodity-1 is as similar to the Commodity-2. It shows that the quickest time to satisfy demand $d_2 = 25$ with proportional capacity sharing is $T = 8$ and with flow-dependent capacity sharing is $T = 7$.

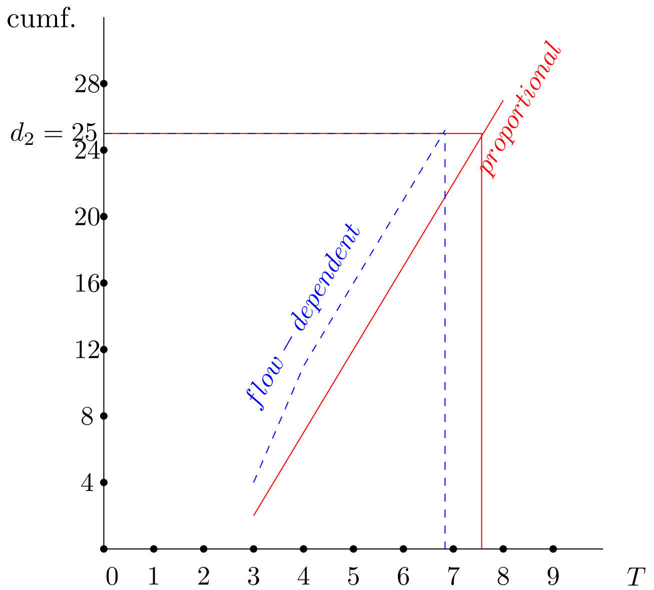

Figure 4.2: Cumulative flow (cumf.) for Commodity-2. Dashed blue curve represents the flow-dependent capacity sharing and solid red curve represents the proportional capacity sharing.

5 Conclusions

Quickest multi-commodity flow (QMCF) problems are relevant to real-world scenarios such as transportation management, communication networks, emergency response, and supply chains, where the goal is to deliver services as quickly as possible. Although QMCF problems are $\mathcal{NP}$-hard, researchers are exploring various approximation methods like length-bounded approximation and condensed time-expanded networks. Not only on fixed transit times but flow-dependent transit times, time-dependent flows, flow-dependent capacity distribution, scaling of time, capacity and cost are active areas of new research.

In this paper, we have presented flow models for the quickest flow problem with multiple commodities and provided an analytical solutions using two different capacity-sharing techniques: proportional and flow-dependent. We developed an algorithm to solve the QMCF problem using a cost-scaling approach with proportional capacity sharing on the bundle arcs, achieving a time complexity of $O(n^3m|K|\log(nC))$. Additionally, we presented an algorithm for solving the problem with flow-dependent capacity sharing by introducing a three-dimensional time-expanded layer graph. Although the time complexity for the quickest flow with flow-dependent capacity sharing is pseudo-polynomial, it achieves quicker solutions for the multi-commodity flow problem than proportional capacity sharing. As far as we know, the solution strategies for the QMCF problem presented in this article are novel contributions. The possible extension of this work can be stochastic demand scenarios or uncertain transit times together with solution strategy using graph condensation.

Conflict of Interest. No.

Data Availability. No additional data is used.

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 for 2021-2023 and University Grants Commission Nepal for PhD Research Fellowship, 2020/21. Similarly, the second author (Shiva Prakash Gupta) thanks to University Grants Commission Nepal for PhD Research Fellowship, 2018/19.
The first, second and fourth authors are deeply shocked to report the untimely demise of the third author, a young and energetic Prof. Dr. Urmila Pyakurel, who passed away on April 12, 2023 at the age of 42. She was a role model Nepalese mathematician with an outstanding research career. As the draft of the paper was finalized with her active involvement, rest of the authors have decided to continue her as a co-author.

References

[1] Ford, L.R. and Fulkerson, D.R., A suggested computation for maximal multi-commodity network flows, *Rand Corporation*, 3 : 27-58 (1958).

[2] Ford, L.R. and Fulkerson, D.R., Flows in networks, *Princeton University Press*, Princeton, New Jersey, (1962).

[3] Khanal, D.P., Pyakurel, U. and Dempe, S., Temporally repeated maximum dynamic flow with intermediate storage, *Operations Research Forum*, (2025). DOI: https://doi.org/10.1007/s43069-025-00474-5.

[4] Pyakurel, U., Khanal, D.P. and Dhamala, T.N., Abstract network flow with intermediate storage for evacuation planning, *European Journal of Operational Research*, 305 : 1178-1193 (2023). DOI: https://doi.org/10.1016/j.ejor.2022.06.054.

[5] Dhamala, T.N., Adhikari, M.C., Khanal, D.P. and Pyakurel, U., Generalized maximum flow over time with intermediate storage, *Annals of Operations Research*, (2024). DOI: https://doi.org/10.1007/s10479-023-05773-w.

[6] Dhamala, T.N., Khanal, D.P. and Dempe, S. , Network restructuring for dynamic flow improvement, *Annals of Operations Research*, (2025). DOI: https://doi.org/10.1007/s10479-025-06496-w

[7] Chen, Y.L. and Chin, Y.H., The quickest path problem, *Computers and Operation Research*, 17 : 153-161 (1990).

[8] Rosen, J.B., Sun, S.-Z. and Xue, G.-L., Algorithms for the quickest path problem and the enumeration of quickest Paths, *Computers and Operations Research*, 18 : 579-584 (1991).

[9] Burkard,R.E., Dlaska, K. and Klinz, B., The quickest flow problem, *ZOR- Methods and Models of Operational Research*, 37 : 31-58 (1993).

[10] Fleischer, L. and Tardos, E., Efficient continuous-time dynamic network flow algorithms, *Operations Research Letter*, 23(3-5) : 71-80 (1998).

[11] Lin, M. and Jaillet, P., On the quickest flow problem in dynamic networks: A parametric min-cost flow approach, *In: Proceeding of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia: Society for Industrial and Applied Mathmatics*, 1343-1356 (2015).

[12] Goldberg, A.V. and Tarjan, R.E., Finding minimum-cost circulations by successive approximation, *Mathematics of Operations Research*, 15(3) : 430-466 (1990).

[13] Ahuja, R. K., Magnanti, T. L. and Orlin, J. B., Network Flows: Theory, Algorithm and Applications, *Prentice Hall, Englewood Cliffs*, Princeton, New Jersey, (1993).

[14] Assad, A., Multi-commodity network flows - a survey, *Networks*, 8(1) : 37-91 (1978).

[15] Iri, M., On an extension of the maximum-flow minimum-cut theorem to multi-commodity flows, *Journal of Operations Research Society of Japan*, 13(3) : 129-135 (1971).

[16] Kennington, J., A survey of linear cost multi-commodity network flows, *Operations Research*, 26 : 209-236 (1978).

[17] Salimifard, K. and Bigharaz, S., The multi-commodity network flow problem: state of the art classification, applications and solution methods,*Springer*, 1-47 (2020).

[18] Wang, I.L., Multi-commodity network flows: A survey part I: applications and formulations, *International Journal of Operations Research*, 15(4) : 145-153 (2018).

[19] Wang, I.L., Multi-commodity network flows: A survey part II: solution methods, *International Journal of Operations Research*, 15(4): 155-173 (2018).

[20] Fan, H., Sun, C., Long, J., Wang, S. and Zeng, E., A novel method for solving the multi-commodity flow problem on evolving networks, *Computer Networks*, 247, (2024). DOI: https://doi.org/10.1016/j.comnet.2024.110451

[21] Lienkamp, B. and Schiffer, M., Column generation for solving large scale multi-commodity flow problems for passenger transportation, *European Journal of Operational Research*, 314(2), 703-717, (2024).

[22] Hall, A., Hippler, S. and Skutella, M., Multi-commodity flows over time: Efficient algorithms and complexity, *Science Direct*, 379 : 387-404 (2007).

[23] Fleischer, L. and Skutella, M., Quickest flows over time, *SIAM Journal on Computing*, 36(6) : 1600-1630 (2007).

[24] Köhler, E. and Skutella, M., Flows over time with load-dependent transit times, *SIAM Journal on Optimization*, 15(4) : 1185-1202 (2005).

[25] Hall, A., Langkau, K. and Skutella, M., An FPTAS for quickest flow multi-commodity flows with inflow-dependent transit times, *Algorithmica*, 47(3) : 299-321 (2007).

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

[27] Khanal, D.P., Pyakurel, U., Dhamala, T.N., Dempe, S. and Schiermeyer, I., Prioritized Maximum Multi-Commodity Flow in Evacuation Planning, *In: Kotsireas, I.S., Nagurney, A., Pardalos, P.M., Pickl, S.W., Vogiatzis, C. (eds) Dynamics of Disasters. Springer Optimization and Its Applications*, vol 217. Springer, Cham (2024). DOI: https://doi.org/10.1007/978-3-031-74006-0_5

[28] Khanal, D.P., Pyakurel, U., Dhamala, T.N. and Dempe, S., Maximum Multi-Commodity Flow with Proportional and Flow-dependent Capacity Sharing, *Comput. Sci. Math. Forum* 2, 5, (2022). DOI: https://doi.org/10.3390/IOCA2021-10904.

[29] Khanal, D.P., Multi-commodity dynamic flow problems with intermediate storage and varying transit times, PhD Thesis, *Tribhuvan University, Institute of Science and Technology*, Kathmandu, Nepal, (2024).