Journal of Mathematics and Statistics

Original Research Paper

Quickest Multi-Commodity Flow Over Time with Partial Lane Reversals

[1]Tanka Nath Dhamala, [2]Shiva Prakash Gupta, [3]Durga Prasad Khanal and [4]Urmila Pyakurel

[1,4]Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal
[2]Tri-Chandra Multiple Campus, Tribhuvan University, Kathmandu, Nepal
[3]Saraswati Multiple Campus, Tribhuvan University, Kathmandu, Nepal

Article history
Received: 21-05-2020
Revised: 29-09-2020
Accepted: 30-09-2020

Corresponding Authors:
Urmila Pyakurel
Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal
Email: urmilapyakurel@gmail.com

Abstract: Routing of more than one different commodity from specific origin nodes to the corresponding destination nodes through the arcs of an underlying network respecting the capacity constraints is one of the main problems in operational research. Among them, the quickest multi-commodity flow problem concerns with minimization of time taken to complete this process. The general multi-commodity and quickest multi-commodity flow problems are computationally hard. By flipping the orientation of lanes towards the demand nodes, the outbound lane capacities are increases. We introduce lane reversals in the quickest multi-commodity flow problem and present two approximation algorithms, one polynomial-time with the help of length-bounded flow and another FPTAS by using Δ-condensed time-expanded graph. Both algorithms prevent reversing arc capacities that are not required by the optimal flows that may be of interest for other purposes.

Keywords: Network Flow, Multi-Commodity, Quickest Flow, Lane Reversals, Length Bounded, Δ-Condensed

Introduction

The multi-commodity network flow problem concerns the distribution of multiple commodities from specific source nodes to corresponding sink nodes obeying the arc capacity constraints. A wide variety of network routing problems can be modeled as a multi-commodity flow problem, for example, message routing in telecommunication, railway network, vehicles routine in transportation, production planning, logistics, supply chains for essential goods, medicines and other supports during disaster and lock-down caused by serious pandemics, like COVID-19. Additional applications and illustrations can be accessed in (Ahuja *et al*., 1993; Assad, 1978; Kennington, 1978; Wang, 2018; Salimifard and Bigharaz, 2020).

The transportation network is considered as a network that corresponds to transshipping different commodities wherein supply points, demand points and the intersection of road segments represent the nodes and links between two nodes denote the arcs (lanes). The initial and final destinations of commodities are at the supply nodes (distribution centers) and demand nodes respectively. The collection of commodities that shipped through the network can be considered as flow. The arcs are assigned with capacity and travel times in the networks with time dimensions.

In general, multi-commodity flow problems can be dichotomized into static and dynamic flow (also known as flow over time) problems. The former one can be classified as maximum, maximum concurrent and minimum cost flow problems. The maximum flow problems arise when the sum of flows for all commodities is to be maximized. The maximum concurrent flow is a special version of the maximum flow problem which maximizes the fraction of demands for all commodities. Whereas, to find the flow value satisfying the demands for all commodities with minimum cost obeying the capacity constraints on all arcs is known as minimum cost flow problem. The dynamic flow problem can be classified into maximum dynamic, quickest and earliest arrival multi-commodity flow problems (Tomlin, 1966; Ali *et al*., 1980; Wang, 2018; Salimifard and Bigharaz, 2020).

The network flow over time has been pioneered six decades ago by Ford and Fulkerson, (1962). The inverse of this problem in which the supply and demand on the origin and destination nodes are given, whereas the problem is to find the minimum possible time to fulfill the demand is called the quickest flow problem.

Burkard *et al*. (1993) applied a binary search to the maximum flow solution of Ford and Fulkerson, (1962) and provided the first polynomial-time bound for the quickest flow problem. They upgraded this method and its complexity to provide strongly polynomial-time bound by incorporating a parametric approach to the minimum cost flow problem.

The quickest flow problem further generalized to multi-source and multi-sink quickest flow problems. One of such generalization, in which the vector of supplies and demands at the terminals are given, where the task is to find a flow over time that satisfies all supplies and demands within minimum possible time, is the quickest transshipment problem. Hoppe and Tardos (2000) presented a polynomial-time solution to this problem.

The naive approach of Ford and Fulkerson (1962) to flow over time problem is generalized to the case of multi-commodities by Fleischer and Skutella (2002; 2007). Moreover, the method proposed by authors employed length-bounded static flows in contrast to the algorithm of Ford and Fulkerson (1962) based one static minimum cost flow computation. Although static flows do not include temporal dimension, Fleischer and Skutella (2007) considered the static flows that suggest accessible routes with reference to transit time as length. They decompose the static flows into paths in which the set of paths starting at some origin node with certain commodity are shipped to its destination node.

Multi-commodity flow problem is harder than their single commodity flow part. Hall *et al*. (2007) proved that multi-commodity flow over time is $\mathcal{NP}$-hard even for series parallel graphs or having only two commodities. The quickest multi-commodity flow problem with or without intermediate node storage and simple flow path is $\mathcal{NP}$-hard. Due to the $\mathcal{NP}$-hardness of this problem, Fleischer and Skutella (2002; 2007) presented two approaches for its approximate solutions. First one is length-bounded flow and the other one is a discretization of larger time step instead of unit time step. They have shown by an instance that the ratio between optimal time horizon without and with storage is 4/3.

To find the quickest time satisfying all demands through storage, Gross and Skutella (2015) focused on speed-up in multi-commodity flow problem. They have shown that there is family of multi-commodity flow instances for which the ratio between the optimal time horizon with and without storage converges to 2. The formulation of the quickest multi-commodity path problem based on mixed-integer linear programming using unique path with uninterrupted scheduling is presented by Melchiori and Sgalambro (2015). Kappmeier (2014) provided solutions of maximum multi-commodity flow over time and multi-source single-sink multi-commodity earliest arrival transshipment problems using a time-expanded network within pseudo-polynomial-time complexity.

Lane reversal means flipping of arc orientations (directions), to increase the flow and reduce the travel time by expanding its capacity. Rebennack *et al*. (2010) provided models and strongly polynomial-time algorithms for two-terminal maximum and quickest flow problems. These lane reversals are made at time zero and kept fixed afterward. Pyakurel and Dhamala (2015; 2017b) find the optimal earliest arrival flow in a two-terminal general network for discrete-time settings. An approximation solution to this problem is given in Pyakurel *et al*. (2017). By using natural transformation Pyakurel and Dhamala (2016; 2017a) provided the solution of quickest flow problem in continuous time setting. Pyakurel *et al*. (2018) presented a strongly polynomial-time algorithm for the quickest flow problem and developed an approximation algorithm with load dependent transit times. Nath *et al*. (2020) investigated the quickest flow problem with an objective of minimum increment of optimal time if some arcs are occupied by facility locations in evacuation network.

The main goal of partial lane reversals is to utilize the capacities of unused arcs in a network for other purposes. Pyakurel *et al*. (2019) introduced the partial lane reversal approach in which only necessary arc capacities are used to increase the flow value. The saved capacities of unused arcs can be used for the logistic supports and facility location in emergency periods.

In this paper, we introduce two approximate algorithms to solve the quickest multi-commodity flow problem with partial lane reversals by reducing them into single-commodity flow problems and decomposing the flow along the paths. Fleischer and Skutella (2002; 2007) approximated the quickest multi-commodity flow problem with the help of *T*-length bounded function and Δ-condensed time-expanded networks. We introduce a partial lane reversal approach in both cases.

By applying lane reversal technique in routing problem, it reduces the time horizon to transship commodities from supply nodes to the demand nodes. The reduction in delivery time is the major significance of this study.

The paper is organized as follows. The first section below provides some basic notations and models used in the article. The quickest multi-commodity flow problem with lane reversals is introduced in the second section. In this section, we present two algorithms for approximate solutions to this problem. The paper is concluded in the last section.

Mathematical Formulations of Flow Models

The multi-commodity flow problem consists of a distribution of different commodities from their respective sources to corresponding sinks through a given network so that the total demand for each commodity is fulfilled. We set necessary denotations and give mathematical formulations of this problem, where reversals of arcs are permitted that improves an objective value by flipping their orientations whenever necessary.

Flow Models for Lane Reversals

Consider a dynamic network topology $\mathscr{N} = (V, A, K, u, \tau, d_i, S, D, T)$ with node set V, arc set $A \subseteq V \times V$ and the set of commodities $K = \{1, 2, ..., k\}$, where $|V| = n$ and $|A| = m$. Each commodity $i \in K$ with demand d_i is routed through a unique source-sink pair (s_i, t_i), where $s_i \in S \subset V$ and $t_i \in D \subset V$. On each arc $e = (v, w)$, the capacity function $u: A \rightarrow \mathscr{R}^+$ restricts the flow of commodities and a non-negative transit time function $\tau : A \rightarrow \mathscr{R}^+$ measures the time to transship the flow from tail (v) to the head (w) of arc $e = (v, w)$. The time period T is given in advanced which is denoted by $\mathscr{T} = \{0, 1, ..., T\text{-}1\}$ in discrete and $\mathscr{T} = [0, T)$ in continuous-time settings.

For a given network $\mathscr{N}$, the corresponding auxiliary network is denoted by $\mathscr{N}^a = (V, A^a, K, u^a, \tau^a, d_i, S, D, T)$, with undirected edges in $A^a = \{(v, w): (v, w) \text{ or } (w, v) \in A\}$, where $e^r = (w, v)$ is the reversed arc of $e = (v, w)$. The capacity of auxiliary arc is the sum of capacities of arcs e and e^r such that $u_e^a = u_e + u_{e^r}$, where $u_e = 0$ if $e \notin A$. The transit time of auxiliary arc τ_e^a is the same as τ_e if $e \in A$ and τ_{e^r} otherwise. Other network parameters are unaltered. Figure 1 (a-b), represents such a transformation for two commodity networks with lane reversals. The first and second commodities are shipped through the paths s_1-t_1 and s_2-t_2, respectively.

The static network without the temporal dimension is also denoted by $\mathscr{N} = (V, A, K, u, d_i, S, D)$. Many nice properties developed based on static network topology are fundamental tools for most of the real-world dynamic flow problems. By considering the auxiliary network $\mathscr{N}^a$ as a two-way directed one, we use it as the network $\mathscr{N}$ in the following models.

Static Multi-Commodity Flow with Lane Reversals

A static multi-commodity flow y with lane reversals for the given static network $\mathscr{N}$ without temporal dimension is a combination of all non-negative static flows y^i with lane reversals denoted by the functions $y^i: A \rightarrow \mathscr{R}^+$ for each commodity i satisfying:

$$\mathscr{G}_y = \begin{cases} d_i & \text{if } v = s_i \\ -d_i & \text{if } v = t_i \quad \forall i \in K \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

$$0 \leq \sum_{i \in K} y_e^i \leq u_e + u_{e^r} \quad \forall e \in A \tag{1.2}$$

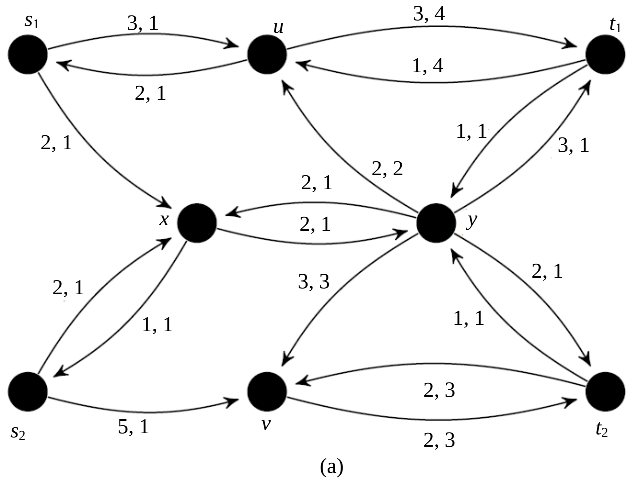
where, the net flow at node v is:

$$\mathscr{G}_y = \sum_{e \in A(v)} y_e^i - \sum_{e \in B(v)} y_e^i$$

The sets $A(v) = \{(v, w) \mid w \in V\}$ and $B(v) = \{(w, v) \mid w \in V\}$ denote outgoing arcs from node v and incoming arcs to node v, respectively, such that $A(D) = \emptyset$ and $B(S) = \emptyset$ except in the lane reversal network. The third condition of the constraints in (1.1) are flow conservation constraints for each commodity at intermediate nodes. The constraints in (1.2) are bundle constraints bounded by lane reversal capacities. The static multi-commodity flow problem with lane reversals seeks to satisfy the pre-specified amount of demand in the remaining conditions of (1.1).

Taking c_e^i as the cost coefficient associated with arc e and commodity i, the cost of static flow y is defined as:

$$c(y) = \sum_{e \in A} \sum_{i \in K} c_e^i y_e^i \tag{1.3}$$



(a)

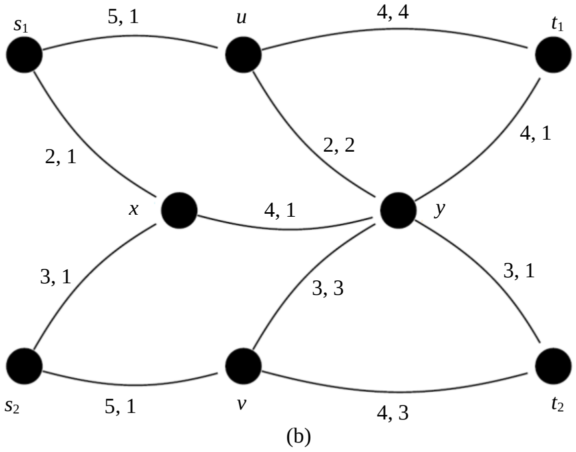


Figure 1: Auxiliary network (b) of the given network (a)

Dynamic Multi-Commodity Flow with Lane Reversals

For a given dynamic network $\mathscr{N}$ with constant transit time on arcs, a multi-commodity flow over times Φ with lane reversals is a collection of flows defined by Φ^i: $A\times\mathscr{T}\rightarrow\mathscr{R}^+$ satisfying the constraints (1.4-1.6):

$$\mathscr{G}_{\Phi}^{T-1}=\begin{cases}d_i & if\ v=s_i\\ -d_i & if\ v=t_i \quad \forall i\in K\\ 0 & otherwise\end{cases} \tag{1.4}$$

$$\mathscr{G}_{\Phi}^{\beta}\le 0\forall v\notin\{s_i,t_i\},i\in K,\beta\in\mathscr{T} \tag{1.5}$$

$$\begin{aligned}&0\le\Phi_e(\theta)=\sum_{i\in K}\Phi_e^i(\theta)\le u_e+u_{e^r}\\ &\forall e\in A,i\in K,\theta\in\mathscr{T}\end{aligned} \tag{1.6}$$

where, the net flow at node *v* at time *δ* is:

$$\mathscr{G}_{\Phi}^{\delta}=\sum_{\theta=0}^{\delta}\sum_{e\in A(v)}\Phi_e^i(\theta)-\sum_{\theta=\tau_e}^{\delta}\sum_{e\in B(v)}\Phi_e^i(\theta-\tau_e)$$

Here, the third condition of the constraints in (1.4) are flow conservation constraints at time horizon *T*, whereas the constraints in (1.5) represent non-conservation of flow at intermediate time points $\beta\in\mathscr{T}$ = {0,1,...,*T*-1}, or with the continuous-time setting, it can be considered as $\mathscr{T}$= [0, *T*). Similarly, the bundle constraints in (1.6) are bounded above by the capacities with lane reversals. The goal is to transship a given amount of flow in order to satisfy the demand d_i of each commodity *i* from s_i to t_i, which is stated in the first two conditions of (1.4). The cost of a discrete flow over time is defined by:

$$c(\Phi)=\sum_{e\in A}\sum_{i\in K}c_e^i\sum_{\theta=0}^{T-1}\Phi_e^i(\theta). \tag{1.7}$$

The bound of cost for every single commodity *i* is calculated as:

$$\sum_{e\in A}c_e^i\sum_{\theta=0}^{T-1}\Phi_e^i(\theta)\le C_i\,, \tag{1.8}$$

and the budget constraints for dynamic flow Φ, stated in (1.7), is bounded by *C*.

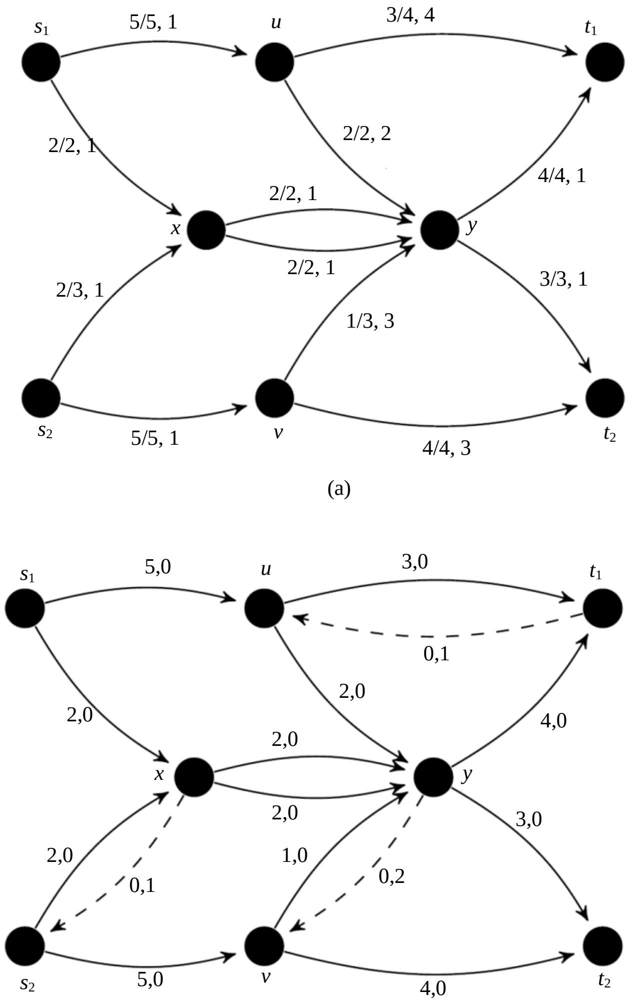
The strict inequality in (1.5) represents weak flow conservation constraints that allow to store the flow at intermediate nodes, wait there for a moment (storage is allowed) and move onwards. The flow over time fulfilling equality constraint in (1.5) represents flow conservation at intermediate nodes for all times *β* without intermediate storage (storage is prohibited). In the case of a single commodity, there is always the quickest flow that does not use intermediate storage. But for the case of multi-commodity, storage is useful. The optimal make-span without storage at intermediate nodes is 4/3 times the optimal make-span with storage, Fleischer and Skutella, (2007).

The single commodity quickest flow problem with lane reversals is modeled as an integer programming problem, (Kim *et al*., 2008). Moreover, they presented a greedy and a bottleneck relief heuristic for its numerical solution. Its single-source and single-sink version is solved in polynomial-time, Rebennack *et al*. (2010). But, the problem with multiple sources and/or sinks is $\mathcal{NP}$-hard, as it is equivalent to 3-SAT and PARTITION, Rebennack *et al*. (2010). The continuous-time version is polynomial-time solvable in (Pyakurel and Dhamala, 2017a; Pyakurel *et al*., 2017). The problem with partial lane reversals is solved in Pyakurel *et al*. (2019).

To realize the main idea behind it, suppose that a single-commodity optimal solution y with partial lane reversals is obtained in the auxiliary network $\mathcal{N}^a$ with respect to a given network $\mathcal{N}$. Then it can be observed from a partial lane reversal strategy that a saving of unused arcs can be achieved in the following ways.

- Arc $e^r = (w, v)$ is reversed if and only if either flow along the arc $e = (v, w)$ is greater than its capacity or there is non-negative flow along the arc $e = (v, w) \notin A$. If $y_e > u_e$ and $u_e^a > y_e$, where $u_e^a = u_e + u_{e^r}$, then the arc e^r is reversed partially and unused capacity of arc e^r is saved.
- If $y_e > u_e$ and $u_e^a = y_e$, then we have to reverse the arc e^r completely. Thus, no capacity is saved.
- If $y_e < u_e$ neither arc e nor arc e^r is reversed, then remaining capacity of arc e and all capacity of arc e^r are saved.

Figure 2(a) illustrates a feasible multi-commodity static flow and Figure 2(b) shows a saving of unused arcs using partial arc reversals. To reallocate the capacity of bundle constraint on arc (x, y), we use the resource-directive decomposition method, where capacity is reallocated in such a way that it increases the objective function value.



(a)

(b)

Figure 2: (a) A static multi-commodity flow and (b) A network with flow on the solid arc and saved capacity on dotted arc.

Discrete Versus Continuous Model

Fleischer and Tardos (1998) connect the relations between discrete and continuous flow models by the notion of natural transformation. It defines the continuous dynamic flow for time $\beta \in [\theta, \theta + 1)$ with $f_e(\beta) = \Phi_e(\theta)$, where $\Phi_e(\theta)$ is the amount of discrete dynamic flow entering arc *e*, at time $\theta \in \{0, 1, ..., T - 1\}$, (Dhamala *et al*., 2018). With the help of this, natural transformation for multi-commodity flow can be extended as follows: Any discrete flow over time $\Phi_e^i(\theta)$ with integral time horizon *T* is equivalent to the continuous flow over time $f_e^i(\beta)$ by incorporating the flow $\Phi_e^i(\theta)$ entering arc *e*, at time step $\theta \le T - \tau_e - 1$ as a constant flow rate on arc *e* during the unit time interval $[\theta, \theta + 1)$.

On the other hand, a continuous flow over time $f_e^i(\theta)$ with integral time horizon *T* and integral transit time τ_e entering arc *e* during the time interval $[\theta, \theta + 1)$ is equivalent to a discrete flow over time $\Phi_e^i(\theta)$ entering arc *e*, at time step θ of the same time horizon *T*, Langkau, (2003). Mathematically, it means that:

$$\Phi_e^i(\theta) = \int_\theta^{\theta+1} f_e^i(\beta)\, d\beta \tag{1.9}$$

for all $e \in A$, $i \in K$ and $0 \le \theta \le T - \tau_e - 1$.

Actually, continuous-time parameters provide comparatively better solutions than their discretization but may suffer from high time complexity. In such a case, discretization is a better option for a good approximation. All the discrete-time algorithms are based on path flows which transship the feasible static flows along the paths.

For standard path decomposition, the static flow y^i of commodity $i \in K$ is decomposed into a set of paths P_i with path flows $\{y_P^i\}_{P \in P_i}$ satisfying $y^i = \sum_{P \in P_i} y_P^i$ and the direction of each arc in s_i-t_i path is in the same direction of y^i. The transit time of each path does not exceed the given time horizon *T*. Feasible dynamic flow is obtained by summing the dynamic flows induced by each of path flows. If feasible flow y^i contains nonstandard path decomposition, then direction of an arc allows being in opposite direction of the flow y^i. In this case, for a path flow y_P^i sending flow along with the arc *e* in the opposite direction of y^i, there must be another path flow $\bar{y}_P^i$ through arc *e* sending the flow in the opposite direction of y_P^i which cancels flow y_P^i on arc *e*.

Naturally, as these transformations can be performed in polynomial-time, the complexity status of discrete and continuous-time solutions remain the same.

Temporally Repeated Multi-Commodity Flow

Let y^i be a static s_i-t_i flow and P_i be a collection of paths carrying flows of commodity $i \in K$ such that $\bigcup_{i=1}^{K} P_i = \mathscr{P}$, where latter is the set of all paths in multi-commodity flow. The family of s_i - t_i paths $P \in P_i$ together with nonnegative flow values y_P^i such that:

$$y_e^i = \sum_{P \in P_i : e \in P} y_P^i \tag{1.10}$$

is called a path decomposition of the flow y^i.

For a feasible static flow y^i in $\mathscr{N}$ with path decomposition $(y_P^i)_{P \in P_i}$ such that the transit time τ_P of every path $P \in P_i$ is bounded from above by *T*, the temporally repeated flow sends flow y_P^i at constant rate in to path $P \in P_i$ starting from time 0 to time $T - \tau_P$.

Lemma 1.1 [*Skutella (2009)]*

Let y^i be a feasible static s_i - t_i flow with its decomposition $(y_P^i)_{P \in P_i}$ such that $y_P^i = 0$ for all $P \in P_i$ with $\tau_P > T$. Then the value of corresponding temporally repeated dynamic flow Φ^i for commodity i is equal to:

$$val(\Phi^i) = \sum_{P \in P_i} (T - \tau_P) y_P^i = T|y^i| - \sum_{e \in A} \tau_e y_e^i. \tag{1.11}$$

Approximation Scheme

Let $\mathscr{X}$ be a minimization (or a maximization) problem. For instance, $I \in \mathscr{X}$, let the optimal solution of the objective function be denoted by OPT (*I*), and let $\varepsilon > 0$. An algorithm *A* is called a (1 + ε) (or (1 − ε)) approximation algorithm for problem $\mathscr{X}$, if for each instance *I* of $\mathscr{X}$ it produces a feasible solution with objective value *A(I)* such that:

$$\begin{aligned} &| A(I) - OPT(I) | \le \varepsilon OPT(I) \\ &\Rightarrow -\varepsilon OPT(I) \le A(I) - OPT(I) \le \varepsilon OPT(I) \\ &\Rightarrow (1-\varepsilon) OPT(I) \le A(I) \le (1+\varepsilon) OPT(I). \end{aligned}$$

- For a problem $\mathscr{X}$, a Polynomial-Time Approximation Scheme (PTAS) is an approximation scheme having time complexity polynomial in the input size of the problem.
- For a problem $\mathscr{X}$, a Fully Polynomial-Time Approximation Scheme (FPTAS) is an approximation scheme having the time complexity polynomial in the input size of the problem and also polynomial in 1/ε.

In many applications, it is more beneficial to calculate an approximate solution quickly than an optimal one. As a consequence, an intense attempt was made to obtain efficient FPTAS for the multi-commodity flow problem.

Quickest Multi-Commodity Flow with Partial Lane Reversals

In this section, we introduce the lane reversal strategy for the quickest multi-commodity flow problem. A solution to this problem satisfies given demands at specified nodes in minimum possible time, where reversals of arcs are permitted. We also develop an efficient algorithm for an approximate solution to this problem. Moreover, our algorithm also applies to save the lanes that are unnecessary to reverse for reducing the time into quickest one. This extends the network flow models introduced in (Fleischer and Skutella, 2007; Hall *et al.*, 2007) into lane reversal framework introduced in (Dhamala *et al.*, 2018; Pyakurel *et al.*, 2019).

Problem 1

Given a network $\mathcal{N} = (V, A, K, u, \tau, d_i, S, D, T)$, the quickest multi-commodity S−D flow problem with partial lane reversals and bounded cost, is to transship the d_i commodity from s_i to t_i for each $i \in K$ in minimum possible time by reversing the direction of necessary arcs at time zero.

The multi-commodity flow over time problem with or without intermediate node storage is $\mathcal{NP}$-hard even in case of series-parallel graphs or having only two commodities, Hall *et al*. (2007). The proof is based on the reduction from the $\mathcal{NP}$-hard PARTITION and 3-PARTITION problems, respectively.

Moreover, the maximum multi-commodity flow problem is $\mathcal{NP}$-hard. Without restriction on intermediate node storage, it can be solved as a static flow problem in the time-expanded graph in pseudo-polynomial-time complexity. The quickest temporally repeated flow with the bounded cost is strongly $\mathcal{NP}$-hard and does not allow FPTAS unless $\mathcal{P} = \mathcal{NP}$. Using 3-PARTITION instances, Hall *et al*. (2007) proved that the quickest multi-commodity flow without intermediate node storage and simple flow paths (paths without loops and cycles) holds the same hardness. Kim *et al*. (2008) proved that lane reversal problem is $\mathcal{NP}$-complete by using 3-SAT. So, the $\mathcal{NP}$-hardness of the quickest multi-commodity flow problem with lane reversals having bounded cost can be stated as follows.

Theorem 2.1

The quickest multi-commodity flow over time problem with partial lane reversals and the bounded cost is $\mathcal{NP}$-hard.

Due to $\mathcal{NP}$-hardness of the quickest multi-commodity flow problem, Fleischer and Skutella (2007) presented two approaches to provide an approximate solution to this problem. The first one is the length-bounded flow and the other one is the discretization of larger time steps instead of unit time steps. We introduce the concept of partial lane reversals in these two cases and provide approximate algorithms to solve Problem 1.

Length Bounded Approximation

A single or multi-commodity path flow wherein each used path must obey a length constraint is a length bounded flow. Let P_i be the set of all s_i - t_i paths in network $\mathcal{N}^a$ with commodity $i \in K$. Then a static flow y is said to be T-length bounded if each component flow y^i for each $i \in K$ can be decomposed into the sum of flows y_P^i i.e., $y^i = \sum_{P \in P_i} y_P^i$ with $y_P^i > 0$ such that the length $\tau_P = \sum_{e \in P} \tau_e$ of any path $P \in P_i$ is at most T. The set of all T-length bounded paths is denoted by $P_i^T = \left\{ P \in P_i : \tau_P \le T \right\} \subseteq P_i$. The problem of finding T-length-bounded static flow y satisfying multi-commodity demands is $\mathcal{NP}$-hard, Fleischer and Skutella (2007). However, an approximated solution to this problem can be computed within polynomial-time.

Lemma 2.2

For any $\varepsilon > 0$, a feasible $(1+\varepsilon)$ T-length bounded static flow y can be computed satisfying given demands d_i with polynomial in the input size and $\frac{1}{\varepsilon}$.

Proof

If we compute a feasible T-length bounded static flow y satisfying the multi-commodity demands d_i, then the lemma will be proved. This flow is obtained by solving the following length-bounded min-cost flow problem, where $c_P^i = \sum_{e \in P} c_e^i$ for all $e \in A^a$ be the cost of a path $P \in P_i$:

$$\min \sum_{i \in K} \sum_{P \in \mathrm{P}_i^T} c_P^i y_P^i \qquad (2.1)$$

$$\textit{such that } \sum_{P \in P_i^T} y_P^i \ge d_i \quad \forall i \in K \qquad (2.2)$$

$$\sum_{i \in K} \sum_{P \in \mathrm{P}_i^T : e \in P} y_P^i \le u_e^a \forall e \in A^a \qquad (2.3)$$

$$y_P^i \ge 0 \ \ \forall i \in K, P \in \mathrm{P}_i^T \qquad (2.4)$$

As the number of paths in P_i^T and the number of variables in the Linear Program (LP) (2.1-2.4) are exponential in the size of the underlying network $\mathcal{N}^a$, the LP problem is converted into dual. Then the LP is reduced to a length-bounded shortest s_i - t_i path with respect to dual arc weights $y_e^i + c_e^i$ having length $\tau(P) \le T$, $P \in P_i^T$. However, the problem is still $\mathcal{NP}$-hard whose approximate solution can be computed as follows. The dual of LP (2.1-2.4) is modified in such a way that for any $\varepsilon > 0$, an s_i - t_i path $P \in P_i$ with $\tau(P) \le (1+\varepsilon)T$ can be found in polynomial-time in the size of $\mathcal{N}^a$ and $\frac{1}{\varepsilon}$ whose length concerning the arc weight $y_e^i + c_e^i$ is at most the length of a shortest path in P_i^T , Fleischer and Skutella, (2007). By solving the modified dual problem with additional constraints, i.e., the length of paths $P \in P_i'$ with $P_i^T \subseteq P_i' \subseteq P_i^{(1+\varepsilon)T}$ is at most $(1+\varepsilon)$ T, an optimal solution can be computed in polynomial-time. From this dual solution, a primal solution y where the flow of commodity i can be sent only on the paths in P_i' is obtained. As $P_i' \subseteq P_i^{(1+\varepsilon)T}$, the obtained flow y is (1+ε) T-length-bounded on the auxiliary network $\mathcal{N}^a$. □

We compute an approximate solution to the quickest multi-commodity flow problem with arc reversals by adopting the concept of length-bounded flows from Fleischer and Skutella (2007) and partial lane reversals from Pyakurel *et al*. (2019), and presenting Algorithm 1.

Algorithm 1 T-length bound quickest multi-commodity flow algorithm with partial lane reversals.
Input: Given a multi-commodity flow network $\mathcal{N}$ = (V, A, K, u, τ, d_i, S, D, T)
Output: The quickest multi-commodity flow with partial lane reversals:

1. The given network is transformed to auxiliary network by adding two-ways capacities in $\mathcal{N}^a$ = (V, A^a, K, u^a, τ^a, d_i, S, D, T) as:

$$u_e^a = \quad u_e + u_{e^r}$$

$$\tau_e^a = \begin{cases} \tau_e & if\ e \in A \\ \tau_{e^r} & otherwise. \end{cases}$$

2. Compute the quickest multi-commodity flow on the transformed network $\mathcal{N}^a$ by using approximate length-bounded algorithm of Fleischer and Skutella, (2007) for quickest multi-commodity flow problem with bounded cost on auxiliary network.
3. Decompose the flow along the s_i - t_i paths and cycles and remove the flows in cycles ∀ $i\in K$.
4. Reverse $e^r \in A$ up to the capacity y_e - u_e , iff $y_e > u_e$, u_e replaced by 0 whenever $e\notin A$, $\forall i$, where y_e $= \sum_{i=1}^{k} y_e^i$ and $u_e = \sum_{i=1}^{k} u_e^i$.
5. For each $e\in A$, if e^r is reversed, $s_c(e^r)$ = $u_e^a - y_e$ and $s_c(e)$ = 0. If neither e nor e^r is reversed, $s_c(e) = u_e - y_e > 0$, where $s_c(e)$ is the saved capacity of e.
6. Transform the solution to the original network.

First, we recall the solution procedure of approximate length-bounded algorithm of Fleischer and Skutella (2007) for the quickest multi-commodity flow problem with bounded cost in Step 2 of our algorithm in Lemma 2.3. Then for the feasibility of Algorithm 1, we prove Lemma 2.4.

Lemma 2.3

The T-length bounded approximate quickest multi-commodity S-D flow with given cost can be computed on an auxiliary network in polynomial-time complexity.

Proof

We have an auxiliary network $\mathcal{N}^a$ with given bounded cost C, predetermined time horizon T and precision $\varepsilon > 0$. Then, a static flow y is computed by averaging the feasible flow over time Φ on each arc as $y_e^i = \frac{1}{T}\sum_0^{T-1}\Phi_e^i$ for all $e\in A^a$ and $i\in K$. The feasibility of y can be shown as follows:

- Capacity constraints:

$$\sum_{i\in K} y_e^i = \frac{1}{T}\sum_{0}^{T-1}\left(\sum_{i\in K}\Phi_e^i\right) \le \frac{1}{T}\sum_{0}^{T-1} u_e^a = u_e^a$$

- Flow conservation:

$$\begin{aligned} &\sum_{e\in A(v)} y_e^i - \sum_{e\in B(v)} y_e^i \\ &= \frac{1}{T}\sum_{0}^{T-1}\left(\sum_{e\in A(v)}\Phi_e^i - \sum_{e\in B(v)}\Phi_e^i\right) \\ &= 0 \forall v \in V \setminus \{s_i, t_i\} \ \ and\ \ i \in K \end{aligned}$$

This flow y satisfies the following conditions: (i) y is (1+ ε) T-length bounded, (ii) y satisfies a fraction $\frac{1}{T}$ of the supplies and demands and (iii) $c(y)$ = $\frac{c(\Phi)}{T}$. We repeat the process until such flow exists. First condition is from Lemma 2.2. As $|y^i| = \frac{1}{T}\sum_0^{T-1}|\Phi^i| \ge \frac{1}{T}d_i$ for all i, so second condition is at hand and third condition is as similar to second one.

By T-length bounded path decomposition of y, flow is now send into each path P with flow rate y_P^i for time horizon T. It takes an additional T units time to reach all the flows at the destination so that total flow will reach at destination in at most 2T time. Thus, arbitrary feasible static flow y satisfying above requirements can be turned into feasible flow over time Φ with time horizon 2T meeting the same requirements as Φ.

For static flow y_e^i, we set cost $c_e = \tau_e$ so that:

$$\sum_{e\in A^a}\tau_e y_e^i = \frac{1}{T}\sum_{e\in A^a}\tau_e\left(\sum_{0}^{T-1}\Phi_e^i\right) = \frac{1}{T}c\left(\Phi^i\right).$$

As each flow is T-length bounded, flow can travel along the path with cost at most T and total cost is bounded above by T |Φ^i|. The temporally repeated flow with time horizon 2T can be calculated as:

$$2T\left|y^i\right| - \sum_{e\in A^a}\tau_e y_e^i = 2\left|\Phi^i\right| - \frac{1}{T}c\left(\Phi^i\right) \ge \left|\Phi^i\right| \ge d_i, \forall i \in K.$$

Thus, static flow y^i satisfies the demand d_i for all $i\in K$ within time 2T.

The length-bounded multi-commodity flow stated above are iteratively send on shortest paths with respect to some length function and computes (1+ε) T-length bounded shortest paths, Fleischer and Skutella (2007).

So, the flow is $(1+\varepsilon)$ T-length bounded. The length boundedness is incorporated to binary search for optimal make-span T^* that yields $T^* \leq T \leq (1+\varepsilon'/4)\ T^*$ for any $\varepsilon' > 0$ to obtain T. By using $\varepsilon = \varepsilon'/4$ we have flow over time with make-span $(2+\varepsilon)\ T \leq (2+\varepsilon')\ T^*$. □

Lemma 2.4

The approximate solution of quickest multi-commodity flow problem with partial lane reversals having bounded cost obtained by Algorithm 1 is feasible.

Proof

To prove the lemma, we have to show that all steps of the Algorithm 1 are well-defined. Step 1 and Step 6 are well-defined transformations. The feasibility of Step 2 is as shown in Fleischer and Skutella (2007). By removing positive flow in cycles in Step 3, flow is either in direction $e = (v, w)$ or in $e^r = (w, v)$ but not in both directions. Hence, the flow obtained from Step 4 is feasible flow with arc reversal in network $\mathcal{N}$. By reversing only necessary arcs in Step 4, Step 5 saves the capacity of unused arcs. Since the obtained flow never exceeds the capacities on each arc in the auxiliary network, reversals of arcs are not conflicting. Hence, all steps are well-defined and so the algorithm is feasible. □

Theorem 2.5

T-length bound approximate solution to the quickest multi-commodity flow problem with partial lane reversals having bounded cost can be obtained by using Algorithm 1.

Proof

A proof of the stated theorem contains two steps, the first one is feasibility and the next one is the optimality of the algorithm. The feasibility of algorithm is achieved in Lemma 2.4.

Now we prove the optimality of the algorithm. From the feasibility, any optimal solution to an approximate quickest multi-commodity flow problem with lane reversals having bounded cost on two-terminal network $\mathcal{N}$ is also a feasible solution to the approximate quickest multi-commodity flow problem with bounded cost on corresponding auxiliary network $\mathcal{N}^a$.

A single source-sink maximum flow problem with lane reversals can be computed in strongly polynomial-time complexity $O(mn+n^2m^3\log n)$, Rebennack *et al.* (2010). By incorporating a parametric search of Burkard *et al.* (1993), a strongly polynomial-time bound for the quickest flow problem with lane reversal can be found. This problem can be solved with partial reversals of lane capacities in time $O(nm^2(\log n)^2)$, Pyakurel *et al.* (2019). On the other hand, the multi-commodity flow problem can be reduced to a single commodity flow problem by separating the paths P_i from s_i - t_i for each commodity $i \in K$ and can be solved as a single commodity flow problem. By Lemma 2.3, an approximate quickest multi-commodity flow solution can be obtained optimally on the auxiliary network $\mathcal{N}^a$. Moreover, any optimal solution on $\mathcal{N}^a$ is equivalent to a feasible solution to given network $\mathcal{N}$.

The unused capacities of the arcs are saved by partial lane reversals. The saved capacity of the arc e^r is $s_c(e^r) = u_e^a - y_e$ if we revert the arc e^r. If neither arc e nor e^r is reversed, then saved capacities of the arcs e and e^r are $s_c(e) = u_e - y_e$ and $s_c(e^r) = u_{e^r}$, respectively. Thus, an approximate quickest multi-commodity flow with partial lane reversals having bounded cost on each arc of the network $\mathcal{N}$ can be computed optimally. □

Corollary 1

A T-length bound approximate quickest multi-commodity flow with partial lane reversals can be computed in polynomial-time complexity.

Proof

The complexity of Algorithm 1 is dominated by Steps 2 and 3. Step 3 is solved in $O(mn)$ time. According to Fleischer and Skutella (2007), Step 2 is solved in polynomial-time. Since other remaining steps can be solved in linear time $O(m)$, the problem can be computed in polynomial-time complexity. □

An FPTAS for the Quickest Multi-Commodity Flow Problem

Consider a multi-terminal dynamic network $\mathcal{N} = (V, A, K, u, \tau, d_i, S, D, T)$ for multi-commodity flows, where all parameters are integers. To get polynomial-time bound, we can rescale the time by large time steps instead of unit time step.

If all transit times on arcs are multiple of $\Delta > 0$ such that $\lceil T/\Delta \rceil$ is bounded by a polynomial in the input size, then the condensed time-expanded network can be obtained by rescaling the time as $\mathcal{N}_T^{\Delta} = \left(V_T^{\Delta}, A_T^{\Delta} \cup A_H^{\Delta}\right)$, where the sets of nodes and edges are defined as:

$$V_T^{\Delta} = \left\{v_{\alpha\Delta} : v \in V, \alpha = 0,1,2,...,\lceil T/\Delta \rceil - 1\right\}$$

$$A_M^{\Delta} = \Big\{\left(v_{\alpha\Delta}, w_{\alpha\Delta+\tau_e}\right) : e = (v,w) \in A,$$
$$\alpha = 0,1,2...,\left\lceil \left(T - \tau_e\right)/\Delta \right\rceil - 1\Big\}$$

$$A_H^{\Delta} = \Big\{\left(v_{\alpha\Delta}, v_{\alpha\Delta+1}\right) : e = \left(v,w\right) \in A,$$
$$\alpha = 0,1,...,\lceil T/\Delta \rceil - 2\Big\}$$

The copies of V_T^{Δ} corresponds to flow through V in time $\mathcal{T} = \{\alpha\Delta\}$ for discrete-time and $\mathcal{T} = [\alpha\Delta, (\alpha+1)\Delta)$ for continuous-time, where $\alpha = \{0, 1, 2, ..., \lceil T/\Delta \rceil - 1\}$. In this setting, for every arc corresponding to a discrete-time with multiple of Δ, capacities are rescaled by Δu_e. If T/Δ is an integer and arc length is defined as above, then any (multi-commodity) flow over time that completes by time T

corresponds to a static (multi-commodity) flow of equal value and cost in $\mathcal{N}_T^\Delta$, Fleischer and Skutella (2007). Similarly, any flow in $\mathcal{N}_T^\Delta$ corresponds to flow over time of equal value that completes by time T. If we take $\Delta = 1$, then Δ-condense time-expanded network reduces to the classical time-expanded network. If arc transit times are not multiple of Δ, then transit times are rounded up to multiple of Δ by $\tau'_e = \lceil \tau_e/\Delta \rceil \Delta$ and $0 \leq \tau'_e - \tau_e < \Delta$ for all arcs $e \in A$.

To develop an FPTAS to solve Problem 1 by using Δ-condensed time-expanded network, we present Algorithm 2. We construct a Δ-condensed auxiliary network wherein capacities of arcs are Δ times the sum of capacities of arcs e and e^r of the given network. Flows are sent through the Δ-condensed auxiliary network $\mathcal{N}^{\Delta a}$ having transit time $\tau_e'^a$. This transformation solves the quickest multi-commodity flow problem with partial lane reversals using FPTAS-Core of Fleischer and Skutella (2007) and the lane reversal technique of Pyakurel *et al*. (2019).

Algorithm 2 An FPTAS for quickest multi-commodity flow problem with partial lane reversals.
Input: Given multi-commodity flow network $\mathcal{N} = (V, A, K, u, \tau, d_i, S, D, T)$.
Output: The quickest multi-commodity flow with partial lane reversals:

1. The auxiliary network $\mathcal{N}^a$ is transformed to Δ−condensed auxiliary network $\mathcal{N}^{\Delta a} = (V, A^a, K, u'^a, \tau'^a, d_i, S, D, T)$ with:
$$u_e'^a = \Delta\left(u_e + u_{e^r}\right)$$
$$\tau_e'^a = \begin{cases} \lceil \tau_e / \Delta \rceil \Delta & if \quad e \in A \\ \lceil \tau_{e^r} / \Delta \rceil \Delta & otherwise. \end{cases}$$
2. Compute the quickest multi-commodity flow with bounded cost on $\mathcal{N}^{\Delta a}$ by using FPTAS-Core of Fleischer and Skutella (2007) with intermediate node storage.
3. Decompose the flow along the s_i - t_i paths and cycles and remove the flows in cycles $\forall i$.
4. Reverse $e^r \in A$ up to the arc capacity y_e - u_e if and only if $y_e > u_e$, u_e replaced by 0 whenever $e \notin A$, $\forall i$, where $y_e = \sum_{i=1}^k y_e^i$ and $u_e = \sum_{i=1}^k u_e^i$.
5. For each $e \in A$, if e^r is reversed, $s_c(e^r) = u_e'^a - y_e$ and $s_c(e) = 0$. If neither e nor e^r is reversed, $s_c(e) = u_e - y_e > 0$, where $s_c(e)$ is the saved capacity of e.
6. Transform the solution to the original network.

Lemma 2.6

An FPTAS for quickest multi-commodity S-D flow problem with bounded cost can be computed on an auxiliary network in fully polynomial-time.

Proof

Consider a single commodity network $\mathcal{N}$ with demand d as input with tentative time horizon T and precision $\varepsilon > 0$. A standard binary search framework is used to calculate bounds on optimal make-span T^*, Fleischer and Skutella (2007). With the help of these bounds, an approximate T with $T^* \leq T \leq (1+O(\varepsilon))\ T^*$ is calculated. By choosing appropriate $\Delta = \varepsilon^2 T/n$, with $T' = \lceil (1+\varepsilon)^3 T/\Delta \rceil \Delta$, there exists a static flow y in the Δ-condensed time-expanded network $\mathcal{N}_{T'}^\Delta$ satisfying demands $(1+\varepsilon)\ d$. With the help of y, a flow over time Φ can be calculated in the network $\mathcal{N}$ with time horizon at most $(1+\varepsilon)\ T'$ satisfying the demands d.

For the correctness of the algorithm, process to find a static flow y in Δ-condensed time-expanded network satisfying demand $(1+\varepsilon)\ d$ at cost $(1+\varepsilon)\ C$ is as follows. As τ'^a and T' are multiple of Δ, any flow over time that completes by time T corresponds to a static flow of equal cost in $\mathcal{N}_{T'}^\Delta$. So, it suffices to show that, there exists a flow over time $\bar{\Phi}$ with time horizon T' satisfying demand $(1+\varepsilon)\ d$ at cost $(1+\varepsilon)\ C$.

For single-commodity flow problems with cost, the storage of flow at intermediate nodes is not essential. So, any flow Φ' on arc e, at time θ can be written in the path flow form as:

$$\Phi_e'(\theta) = \sum_{P \in P : e \in P} \Phi_P'\left(\theta - \tau_e(P)\right) \tag{2.5}$$

where, $\tau_e(P)$ is the length of sub-path obtained by removing arc e and its successors. This flow is reduced to a smooth flow over time by:

$$\hat{\Phi}_P(\theta) = \frac{1}{\varepsilon T} \sum_{\zeta = \theta - \varepsilon T}^{\theta} \Phi_P'(\zeta) \tag{2.6}$$

for all $\theta \in \{0, ..., (1+\varepsilon)^2\ T + \varepsilon\ T\}$. Finally, taking $\bar{\Phi} = \hat{\Phi}/(1+\varepsilon)$, flow $\bar{\Phi}$ is required flow satisfying demand $(1+\varepsilon)d$ at cost $(1+\varepsilon)C$ and time at most T' which corresponds to the static flow y. Similarly, for a given flow y, we can compute a flow over time Φ in $\mathcal{N}$ satisfying demand d, cost bounded by C and time horizon bounded by $(1+\varepsilon)\ T'$. Hence, an approximate solution of quickest flow problem with bounded cost can be obtained from $O\left(\log \frac{1}{\varepsilon}\right)$ static min cost flow computations in $\mathcal{N}_T^\Delta$ with $O\left(\frac{n^2}{\varepsilon^2}\right)$ nodes and $O\left(\frac{mn}{\varepsilon^2}\right)$ arcs. By using the storage of flow at intermediate nodes this result is generalized to the case of multi-commodity flow. The idea is that, if there is intermediate storage, there may be flow which contains non-simple paths that can be reduced to a simple path by delaying for some time σ.

To deal this, in the path-based model, a path P^σ with delays σ_j, $j = 1, 2, ..., q$ is a path on network $\mathcal{N}$ having nodes

$(v_0, v_1, ..., v_q)\in P^\sigma$. The amount of time wherein flow is stored at node v_j while moving forward from node v_j to node v_{j+1} on path P^σ particularize by delay σ_j, $j = 1, 2, ..., q$. The flow over time Φ_{P^σ} is the decomposition of flow over time Φ on paths with delay P^σ by:

$$\Phi_e(\theta)=\sum_{P^\sigma:e\in P}\Phi_{P^\sigma}\left(\theta-\tau_e\left(P^\sigma\right)\right) \qquad (2.7)$$

where, $\tau_e(P^\sigma)$ is the time with delays on path P^σ to reach the arc e.

In each search step of the algorithm, we check whether $\mathcal{N}_T^{\Delta}$ contains quickest multi-commodity flow satisfying all demands d_i having bounded cost with intermediate node storage and update the value of T. The lower bound can be computed in polynomial-time using constant factor approximation algorithm. The estimated T' can thus be found within $O\left(\log\frac{1}{\varepsilon}\right)$ geometric mean binary search steps by static multi-commodity flow computations with bounded cost having the same number of nodes and arcs. □

Lemma 2.7

The solution of the quickest multi-commodity flow problem with partial lane reversal having bounded cost obtained by Algorithm 2 is feasible.

Proof

Being transformations, Step 1 and Step 6 are feasible. The feasibility of Step 2 is shown in Lemma 2.6 and the feasibility of Step 4 is similar to the feasibility of Step 4 of Lemma 2.4. Step 3 is already feasible. Step 5 saves the unused condensed capacity of the arc obtained by Step 4. Hence all the steps of Algorithm 2 are well-defined. Thus, Algorithm 2 gives a feasible solution. □

Theorem 2.8

An FPTAS provides an approximate solution to the quickest multi-commodity flow problem with partial lane reversals having bounded cost by using Algorithm 2.

Proof

We prove this theorem in two steps. In the first step, Lemma 2.7 proves the feasibility. In the next step, we show that the obtained solution is optimal. Feasibility implies that an optimal solution to an approximate quickest multi-commodity flow problem on two-terminal network $\mathcal{N}$ with lane reversals having bounded cost is also a feasible solution to the approximate quickest multi-commodity flow with bounded cost on the corresponding auxiliary network $\mathcal{N}^{\Delta a}$. A multi-commodity flow over time can be reduced to a static flow problem on time-expanded network and is computed in pseudo-polynomial-time complexity, Skutella (2009). The Δ-condensed time-expanded network as defined above is time-expanded network obtained by reduction of its size which provides approximate polynomial-time bound. An approximate quickest flow solution can be obtained optimally on auxiliary network by Lemma 2.6. The relation of flow over time to quickest flow problem and reduction of multi-commodity to single commodity are similar as in Theorem 2.5. Furthermore, any optimal solution on $\mathcal{N}^{\Delta a}$ is equivalent to the feasible solution to given network $\mathcal{N}$. The unused capacities of the arcs by partial lane reversals are saved in Step 5. Consequently, an approximate quickest multi-commodity flow solution with lane reversals having bounded cost on each arc of a given network $\mathcal{N}$ can be computed optimally. □

Corollary 2

An approximate quickest multi-commodity flow with partial lane reversal can be computed in fully polynomial-time complexity.

Proof

The complexity of Algorithm 2 is dominated by Steps 2 and 3. Step 3 is solved in $O(mn)$ time. According to Fleischer and Skutella (2007), a $(1+\varepsilon)$ approximate solution of static multi-commodity flow problem with bounded cost can be calculated by $O\left(\log\frac{1}{\varepsilon}\right)$ computation in a $\mathcal{N}_T^{\Delta}$ with $\left(\frac{n^2}{\varepsilon^2}\right)$ nodes and $\left(\frac{mn}{\varepsilon^2}\right)$ arcs in Step 2. Since other remaining steps can be solved in linear time $O(m)$, so the problem can be computed in fully polynomial-time. □

Example 1

Consider the networks from Figure (1-3) with two commodities having demands $d_1 = d_2 = 12$. Figure 1(a) and 2(a) are used to calculate the quickest time without lane reversals and with lane reversals, respectively. Similarly, Figure 3(b) represents the Δ-condensed network of Figure 3(a) for $i = 1$ by taking $\Delta = 2$ with modified capacities and transit times on the arcs. The approximate solution by Δ-condensed time-expanded network is presented in Figure 3(c). The same procedure can be used for the second commodity.

On solving Problem 1 for quickest multi-commodity flow over time without lane reversals (c.f. Figure 1(a)), it takes $T = 7$ units of time to fulfill the given demands d_1 and d_2. On the other hand, if lane reversal is applied (c.f. Figure 2(a)), then it takes $T = 5$ units of time to fulfill the same demands. So, approximately 28.5% of the time is saved due to lane reversal.

Again, to use length-bounded approximation without lane reversal (c.f. Figure 1(a)), 5-length bounded approximation is essential for the first commodity to satisfy the demand $d_1 = 12$ and flow is pushed for 3 times, i.e., θ = 0, 1, 2. So the quickest time for the first commodity is T = 7. Similarly, 4-length bounded approximation is sufficient for the second commodity to satisfy the demand d_2 = 12, and flow is pushed for 4-time units so that the quickest time for the second commodity is T = 7. Therefore, the minimum time required to satisfy both demands is T = 7. But in Figure 2(a) with lane reversals, 4-length bounded approximation is sufficient for the first commodity, and flow is pushed for 3-time units whereas 4-length bounded approximation is sufficient for the second commodity and flow is pushed for 2-time units. The quickest time to satisfy both demands is T = 6. Thus approximately 14.3% time is saved due to lane reversal in length-bounded approximation.

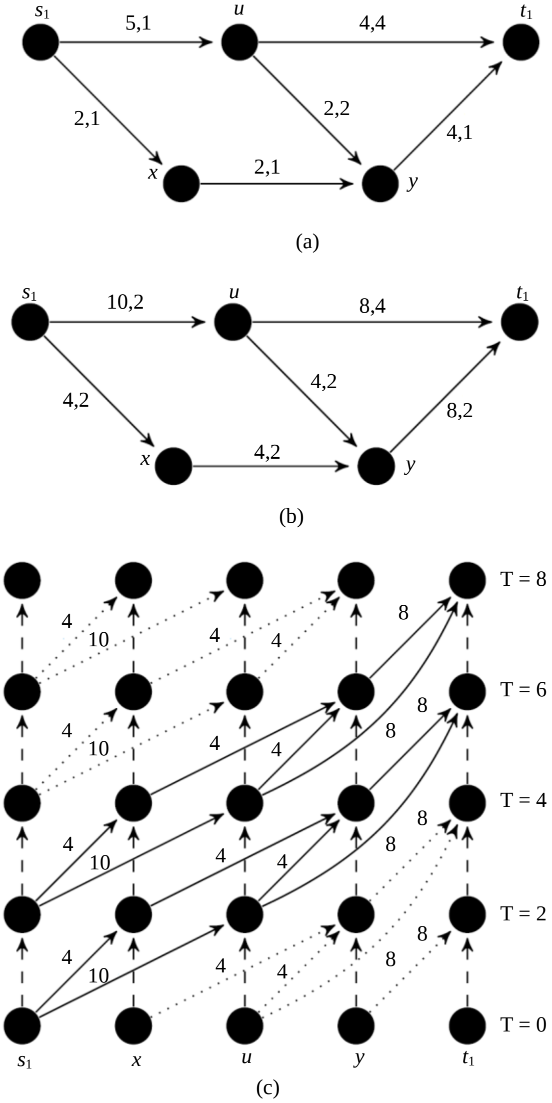
Next, by using Δ-condensed time-extended network presented in Figure 3(c), the total time to fulfill demand d_1 with lane reversal is T = 6. Similarly, we can calculate the quickest time for d_2 and the total time to satisfy both demands is T = 6. If we calculate flow in Δ-condensed time-extended network without lane reversals (c.f. Figure 1(a)) with Δ = 2, then T = 8 time is required to fulfill the given demands. It shows that lane reversal configuration improves the quickest time by 25% even in Δ-condensed network.

We summarize the Example 1 from the following Table 1.

Table 1: Quickest time with and without lane reversals

Case	Without LR	With LR	%Change
Normal	7	5	28.5
Length bound	7	6	14.3
Δ-condensed	8	6	25

LR = Lane Reversals



(a)

(b)

(c)

Figure 3: (a) Network having capacity, transit time on the arc for first commodity after lane reversals. (b) Δ-condensed network with condensed capacity and condensed transit time of (a) with Δ = 2. (c) Δ-condensed time expanded network of (b).

Conclusion

Routing of several commodities from the origin to the destination through a network is one of the main problems in operations research. The minimization of time (cost) is a vital issue. To fulfill the demands in minimum possible time, a well-known quickest flow problem has been investigated. The quickest flow problem for a single commodity can be solved efficiently, but the multi-commodity case is weakly $\mathscr{NP}$-hard. However, a polynomial-time approximation algorithm by length-bounded function and an FPTAS-Core by using Δ-condensed time-expanded network have been obtained.

To improve the quickest time in the two-way network a lane reversals strategy is an important tool. We incorporate this technique for length-bounded approximation as well as condensed time-expanded network.

In this paper, we investigated the quickest multi-commodity flow problem with partial lane reversals. We introduced its mathematical model and presented two algorithms, one a polynomial-time approximation and another an FPTAS. As we have studied the quickest multi-commodity flow problem with constant transit time, we are further interested in elongate these approaches for flow-dependent attributes. The results obtained in this study are both theoretical and practical interests.

This work will be helpful to the researchers who intended to extend their ideas for time-dependent, flow-dependent and load-dependent attributes.

Acknowledgement

The second author thanks the University Grants Commission, Nepal, for the Ph.D. research fellowship. The fourth author thanks Alexander von Humboldt Foundation for the Return Fellowship (November 2019-October 2020). The authors would also like to thank the anonymous referees and the editor for their valuable suggestions to improve the quality of this paper.

Author’s Contributions

Tanka Nath Dhamala: Supervision.

Urmila Pyakurel: Conceptualization, editing and supervision.

Shiva Prakash Gupta and Durga Prasad Khanal: Formal analysis, investigation and documentation.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network flows: Theory, algorithms and applications. Prentice-Hall.

Ali, A., Helgason, R., Kennington, J., & Lall, H. (1980). Computational comparison among three multicommodity network flow algorithms. Operations Research, 28(4), 995-1000.

Assad, A. (1978). Multicommodity network flows-a survey. Networks, 8(1), 37-91.

Burkard, R. E., Dlaska, K., & Klinz, B. (1993). The quickest flow problem. Zeitschrift für Operations Research, 37(1), 31-58.

Dhamala, T. N., Pyakurel, U., & Dempe, S. (2018). A critical survey on the network optimization algorithms for evacuation planning problems. International journal of operations research, 15(3), 101-133.

Fleischer, L., & Skutella, M. (2002, May). The quickest multicommodity flow problem. In International Conference on Integer Programming and Combinatorial Optimization (pp. 36-53). Springer, Berlin, Heidelberg.

Fleischer, L., & Skutella, M. (2007). Quickest flows over time. SIAM Journal on Computing, 36(6), 1600-1630.

Fleischer, L., & Tardos, É. (1998). Efficient continuous-time dynamic network flow algorithms. Operations Research Letters, 23(3-5), 71-80.

Ford, L. R., & Fulkerson, D. R. (1962). Flows in networks. Princeton university press, Princeton, New Jersey. 276, 22.

Gross, M., & Skutella, M. (2015). A tight bound on the speed-up through storage for quickest multi-commodity flows. Operations Research Letters, 43(1), 93-95.

Hall, A., Hippler, S., & Skutella, M. (2007). Multicommodity flows over time: Efficient algorithms and complexity. Theoretical computer science, 379(3), 387-404.

Hoppe, B., & Tardos, É. (2000). The quickest transshipment problem. Mathematics of Operations Research, 25(1), 36-62.

Kappmeier, J. P. (2014). Generalizations of flow over time with applications in evacuation optimization. Ph.D. thesis, Technical university, Berlin.

Kennington, J. L. (1978). A survey of linear cost multicommodity network flows. Operations Research, 26(2), 209-236.

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

Langkau, K. (2003). Flows over time with flow-dependent transit times. Ph.D. thesis, Technical university, Berlin.

Melchiori, A., & Sgalambro, A. (2015). Optimizing emergency transportation through multicommodity quickest paths. Transportation Research Procedia, 10, 756-765.

Nath, H. N., Pyakurel, U., Dhamala, T. N., & Dempe, S. (2020). Dynamic network flow location models and algorithms for quickest evacuation planning. Journal of Industrial & Management Optimization, doi: 10.3934/gimo.2020102.

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

Pyakurel, U., & Dhamala, T. N. (2016). Continuous time dynamic contraflow models and algorithms. Advances in Operations Research, 2016.

Pyakurel, U., & Dhamala, T. N. (2017a). Continuous dynamic contraflow approach for evacuation planning. Annals of Operations Research, 253(1), 573-598.

Pyakurel, U., & Dhamala, T. N. (2017b). Evacuation planning by earliest arrival contraflow. Journal of Industrial & Management Optimization, 13(1), 487-501.

Pyakurel, U., Dhamala, T. N., & Dempe, S. (2017). Efficient continuous contraflow algorithms for evacuation planning problems. Annals of Operations Research, 254(1-2), 335-364.

Pyakurel, U., Nath, H. N., & Dhamala, T. N. (2018). Efficient contraflow algorithms for quickest evacuation planning. Science China Mathematics, 61(11), 2079-2100.

Pyakurel, U., Nath, H. N., Dempe, S., & Dhamala, T. N. (2019). Efficient dynamic flow algorithms for evacuation planning problems with partial lane reversal. Mathematics, 7(10), 1-29.

Rebennack, S., Arulselvan, A., Elefteriadou, L., & Pardalos, P. M. (2010). Complexity analysis for maximum flow problems with arc reversals. Journal of Combinatorial Optimization, 19(2), 200-216.

Salimifard, K., & Bigharaz, S. (2020). The multicommodity network flow problem: State of the art classification, applications and solution methods. Operational Research, 1-47.

Skutella, M. (2009). An introduction to network flows over time. In Research trends in combinatorial optimization (pp. 451-482). Springer, Berlin, Heidelberg.

Tomlin, J. A. (1966). Minimum-cost multicommodity network flows. Operations Research, 14(1), 45-51.

Wang, I. L. (2018). Multicommodity network flows: A survey, Part I: Applications and Formulations. International Journal of Operations Research, 15(4), 145-153.