DYNAMIC NETWORK CONTRAFLOW EVACUATION PLANNING PROBLEM

A THESIS SUBMITTED TO THE

CENTRAL DEPARTMENT OF MATHEMATICS
INSTITUTE OF SCIENCE AND TECHNOLOGY
TRIBHUVAN UNIVERSITY, NEPAL

FOR THE AWARD OF
DOCTOR OF PHILOSOPHY
IN MATHEMATICS

BY

PHANINDRA PRASAD BHANDARI

JUNE 2020

DYNAMIC NETWORK CONTRAFLOW EVACUATION PLANNING PROBLEM

A THESIS SUBMITTED TO THE

CENTRAL DEPARTMENT OF MATHEMATICS
INSTITUTE OF SCIENCE AND TECHNOLOGY
TRIBHUVAN UNIVERSITY, NEPAL

FOR THE AWARD OF
DOCTOR OF PHILOSOPHY
IN MATHEMATICS

BY

PHANINDRA PRASAD BHANDARI

JUNE 2020

DECLARATION

Thesis entitled **"Dynamic Network Contraflow Evacuation Planning Problem"** which is being submitted to the Central Department of Mathematics, Institute of Science and Technology(IOST), Tribhuvan University, Nepal for the award of the degree of Doctor of Philosophy (Ph.D.), is a research work carried out by me under the supervision of Assoc. Prof. Dr. Shree Ram Khadka, Central Department of Mathematics, Tribhuvan University, Nepal and co-supervised by Prof. Dr. Stefan Ruzika, Department of Mathematics, Technische Universität Kaiserslautern, Germany.

This research is original and has not been submitted earlier in part or full in this or any other form to any university or institute, here or elsewhere, for the award of any degree.

....
Phanindra Prasad Bhandari

RECOMMENDATION

This is to recommend that **Phanindra Prasad Bhandari** has carried out research entitled **"Dynamic Network Contraflow Evacuation Planning Problem"** for the award of Doctor of Philosophy (Ph.D.) in **Mathematics** under our supervision. To our knowledge, this work has not been submitted for any other degree.

He has fulfilled all the requirements laid down by the Institute of Science and Technology (IOST), Tribhuvan University, Kirtipur for the submission of the thesis for the award of Ph.D. degree.

....

Shree Ram Khadka, Ph.D.
Supervisor
(Associate Professor)
Central Department of Mathematics, Tribhuvan University
Kirtipur, Kathmandu, Nepal

....

Stefan Ruzika, Ph.D.
Co-Supervisor
(Professor)
Department of Mathematics, Technische Universität Kaiserslautern
P.O. Box 3049, 67663 Kaiserslautern, Germany

June 2020

TRIBHUVAN UNIVERSITY

CENTRAL DEPARTMENT OF MATHEMATICS

OFFICE OF THE HEAD OF DEPARTMENT

KIRTIPUR, KATHMANDU
NEPAL

Ref. No.:

LETTER OF APPROVAL

Date: 14/07/2020

On the recommendation of Assoc. Prof. Dr. Shree Ram Khadka and Prof. Dr. Stefan Ruzika, this Ph.D. thesis submitted by Phanindra Prasad Bhandari, entitled "**Dynamic Network Contraflow Evacuation Planning Problem**" is forwarded by Central Department Research Committee (CDRC) to the Dean, IOST, T.U..

....

Prof. Dr. Tanka Nath Dhamala
Head,
Central Department of Mathematics, Tribhuvan University
Kirtipur, Kathmandu, Nepal

Telephone No.: 4331977, E-mail: cdmath@wlink.com.np

ACKNOWLEDGEMENTS

Foremost, I would like to express my deep and sincere gratitude to my supervisor Assoc. Prof. Shree Ram Khadka who introduced me to Optimization and trusted to conduct this Ph.D. research. His continuous guidance, encouragement with insightful comments and immense knowledge helped me all the time of research. This thesis would hardly exist without his contribution.

I express my warmest gratitude to my co-supervisor Prof. Stefan Ruzika for his guidance and support. I express my sincere thanks to Prof. Tanka Nath Dhamala, head of Central Department of Mathematics for his valuable scientific support throughout my research period. I thankfully acknowledge Luca E. Schäfer for his academic help.

I express my sincere gratitude to UGC Nepal for awarding me with the Ph.D. fellowship (Award No.: PhD-72/73-S&T-01). I am also grateful to the support of DAAD Partnership Program in GraThO project at Central Department of Mathematics, Tribhuvan University and to the Nepal Mathematical Society for providing the NMS-Fellowship award. I would like to owe my warm gratitude to Khwopa Engineering College, Nepal and the Principal Sujan Maka for the support in various forms.

I have greatly profited from encouraging suggestions of professors, other faculties and colleagues at the department. Efficient services provided by staffs at the department as well as at the Dean's office made administrative procedures smooth throughout the research duration.

I am forever indebted to my late parents Bishnu Prasad and Tulasa Devi for their unconditional sacrifices that have made me who I am.

Finally, I express my heart-felt love to my wife Shanti Sharma Bashyal who encouraged me to take this step and took almost all hardships of our family life. My thanks with endless love goes to our little boys Anwit and Aruj for making me stronger and more fulfilled than I could have ever imagined. Likewise, I thank to all those who wish a success.

Phanindra Prasad Bhandari
June 2020

ABSTRACT

Evacuation planning problem gives efficient way-out on existing road network that attempts to shift evacuees from risk zone to safer in minimum time with minimum casualty during disasters. Its domain based on network flow problems has been flourished with models and solutions with various network attributes. A common feature on almost all of these models is that the flow function obeys conservation constraints at each intermediate vertex. In particular, maximum dynamic flow (MDF) problem, earliest arrival flow (EAF) problem and quickest flow (QF) problem have great applicability in evacuation planning problems.

Contraflow approach reconfigures the network identifying ideal direction and reallocating available capacity for each arc to improve flow egress time and/or improve the number of flow units from source to sink. This thesis sketches a brief survey of models and results on contraflow evacuation planning problems. Continuous time model for maximum dynamic contraflow (MDCF) problem is studied with its efficient solution. Thesis also extends contraflow model for multi-network. Network modification strategy is applied to give polynomial time algorithms to solve the problems; namely, MDCF problem and earliest arrival contraflow (EACF) problem based on extended model with discrete as well as continuous time setting. The former problems are considered in general networks whereas the latter problems in two terminal series parallel (TTSP) networks. Arc reversibility is allowed only once at time zero in each of the cases.

Evacuation models with intermediate temporary shelters could be extra benefit while implementing them. This thesis formulates, as another contribution, flow model for network with capacitated vertices of given priority order in which flow conservation may be violated. This violation makes possible for flow units to be held at intermediate vertices which turns out to be applicable in modeling an evacuation planning problem with intermediate holding of evacuees at temporary shelters despite sending them into the sink. Based on this model, maximum flow problem is considered and proposed a polynomial solution for static case and pseudo-polynomial solution for dynamic case. Also, polynomial solutions for MDF problem and QF problem modeled on uniform path length (UPL) network and for EAF problem modeled on UPL-TTSP network are proposed. As the final contribution, contraflow approach is linked to evacuation problems with capacitated prioritized vertices.

Keywords: Network flow models, Contraflow, Capacitated vertices, Evacuation planning problem, Disaster management.

LIST OF ABBREVIATIONS

UPL	uniform path length
TTSP	two terminal series parallel
MSF	maximum static flow
MDF	maximum dynamic flow
QF	quickest flow
EAF	earliest arrival flow
MSCF	maximum static contraflow
MSTCF	maximum static transshipment contraflow
MDCF	maximum dynamic contraflow
QCF	quickest contraflow
EACF	earliest arrival contraflow
DT-MDCF	discrete time maximum dynamic contraflow
CT-MDCF	continuous time maximum dynamic contraflow
DT-EACF	discrete time earliest arrival contraflow
CT-EACF	continuous time earliest arrival contraflow
DT-QCF	discrete time quickest contraflow
LexMSF	lexicographic maximum static flow
DT-LexMDF	discrete time lexicographic maximum dynamic flow
LexQF	lexicographic quickest flow
DT-LexEAF	discrete time lexicographic earliest arrival flow
LexMSCF	lexicographic maximum static contraflow
DT-LexMDCF	discrete time lexicographic maximum dynamic contraflow
DT-LexEACF	discrete time lexicographic earliest arrival contraflow

LIST OF SYMBOLS

G	graph
$\mathcal{N}$	network
v	vertex
V	set of vertices
s	source vertex
s^*	super source
d	sink vertex
d^*	super sink
$\mathcal{S}$	set of terminals (sink and intermediate vertices)
$\mathcal{S}^-$	set of sinks
$\mathcal{S}^+$	set of sources
$k(v)$	capacity at vertex v
a	arc
A	set of arcs
$c(a)$	cost for unit flow on arc a
$l(a)$	lower capacity of arc a / demand at arc a
$u(a)$	upper capacity of arc a
$\tau(a)$	transit time of arc a
$\mu(v)$	demand at vertex v
$C(\mu(v))$	total cost for shipping demand $\mu(v)$
$f(a)$	flow along arc a
t	time
T	fixed time horizon
f_m	minimum cost circulation on network
f_s	static flow
$\mathbf{f}_s$	static flow value

f_d	discrete dynamic flow
$\mathbf{f}_d$	discrete dynamic flow value
f_c	continuous dynamic flow
$\mathbf{f}_c$	continuous dynamic flow value
$\mathcal{C}$	cycle on network
m	number of arcs
n	number of vertices
$\mathbb{N}$	set of positive integers
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathcal{N}^T$	time expanded network for time horizon T
$\mathcal{N}^+$	network after inserting artificial vertex to each arc on $\mathcal{N}$
$\mathcal{N}'$	network after converting vertex capacities on $\mathcal{N}$ to arc capacities
$\mathcal{N}^{inv}$	inverse network of $\mathcal{N}$
$\tilde{\mathcal{N}}$	transformed network of $\mathcal{N}$ for contraflow purpose
$\mathcal{N}_f$	residual network of $\mathcal{N}$ with respect to flow f
$\mathcal{N}^*$	network consisting of arcs $a \in A$ having a positive flow $f(a)$ only
A_f	set of residual arcs on $\mathcal{N}_f$
$\mathcal{N}_{\Gamma_{v_k}}$	residual network of $\mathcal{N}$ after solving the LexMCC problem on $\mathcal{N}$ with $\lvert\mathcal{S}\rvert = k$
$u_r(a)$	residual capacity of arc a
$k_r(v)$	residual capacity of vertex v
h	minimum residual capacity of a path
C	maximum arc cost
U	maximum arc capacity
(v, w)	arc joining vertex v and w with v the tail and w the head
$s - d$ path	directed path from source s to sink d in $\mathcal{N}$
γ	$s - d$ path
$s - v_i$ path	directed path from source s to vertex v_i in $\mathcal{N}$
γ_{v_i}	$s - v_i$ path
Γ	set of paths γ
$f(\gamma)$	flow value that can travel at a time along γ
$\tau(\gamma)$	transit time of path γ
$c(\gamma)$	cost of path γ
Γ^T	temporally repeated flows of paths in Γ within time horizon T

Γ_{v_i}	set of paths γ_{v_i}
$\mathcal{T}$	$\{0, 1, \ldots, \mathrm{T}\}$
$\Gamma_{v_i}^E$	extended set of paths γ_{v_i} that exist at any time step $t \in \mathcal{T}$
$N(\gamma_{v_i})$	actual number of times that the flow along γ_{v_i} is repeated
$I_t(\gamma_{v_i})$	the time step at which the flow along γ_{v_i} starts to get repeated
$F_t(\gamma_{v_i})$	the time step after which the flow along γ_{v_i} stops to get repeated
$\gamma_{v_i^p}$	p^{th} path in $\Gamma_{v_i}^E$
$ex_f(v)$	excess flow at vertex v
$ex_f(v,t)$	excess flow at vertex v at time t

LIST OF FIGURES

LIST OF ALGORITHMS

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Increasing number of disasters due to different kinds of natural and human caused hazards have drawn attention of stakeholders (academicians, researchers, authorities, etc.) for preparation of optimal evacuation plans. Such plan directs how people at risk zone(s) can be shifted to pre-specified safer zone(s) within a reasonable time bound along existing road network topology of evacuation region. Evacuations could be of a region triggered by earthquakes, floods, tsunamis or hurricanes, of airplanes having engine turbine problems, of buildings on fire or bomb threat, etc. By transforming specific places and junctions into vertices and accesses between them (bridges or links) into arcs, it is intuitive to see an evacuation region as a network prototype. The movement of evacuees on roads behaves like a flow passing through arcs. Besides numerous simulation tools (microscopic approach), mathematical optimization (macroscopic approach) provides important tools for modeling, preparing and managing evacuation tasks. Macroscopic models are used to produce a considerably good lower bounds for evacuation time and do not consider any individual behavior during the emergency situation, whereas microscopic models are able to model individual behavior and interactions among evacuees which influence their movement [22, 50, 52]. Macroscopic evacuation planning problems can be modeled as network optimization flow problems. Studying network flow problems and designing implementable algorithms for solving them are crucial due to their intuitive applicability in expressing a wide variety of real world problems, including evacuation planning problems.

A network consists of vertices and arcs. To each arc, capacity is assigned which bounds the amount of flow that can travel along it. For evacuation setting, vertices are categorized as source vertices, sink vertices and intermediate vertices denoting risk zones with evacuees, safe zones with open spaces and basic facilities and transshipment vertices, respectively. In some special cases, intermediate vertices also have transshipment capacities and storage capacities. A static network is not capable of capturing the detail of evacuation scenario. In dynamic networks, in addition to the arc capacity denoting the number of lanes, each arc has a transit time that is required for evacuee to travel the arc. If time is taken in a discrete manner, the capacity of an arc is amount of flow that may enter the arc at each time step. While modeling evacuations, time being continuous variable, it is quite natural to consider the transit times in continuous manner in which capacity of an arc is the rate of flow that can enter the arc. Transit times could also be congestion dependent in some real world evacuation problems. This aspect is beyond the boundary of this thesis. However, we consider time variable in both discrete as well as continuous time settings. A detailed description of network structure and parameters imposed on it with mathematical denotations has been presented in Section 3.2.

Maximum dynamic flow (MDF) problem (see Subsection 3.5.1 for details), introduced by Ford and Fulkerson [42, 43], is a central problem in network flow theory that attempts to determine maximum flow of single commodity from a single source to a single sink over specified time horizon. The problem and its applications have been extensively studied in the literature, e.g., [21], [29], [47], etc. Dynamic extensions of these problems often provide important features for modeling real-world applications, e.g., in evacuation scenarios. Many dynamic network flow problems have been investigated in the context of evacuation planning problems in literature, e.g., [7], [8], [22], [53], [64], [87], [89], etc. We focus on the extensions of some evacuation planning models based on MDF problem by imposing two different characteristics: contraflow, i.e., arc reversal capability (in Chapter 4) and intermediate holding capability, i.e., relaxation of flow conservation constraints and consideration of capacitated vertices (in Chapter 5).

The dynamic flow problem closely related to MDF problem is *quickest flow (QF) problem* that asks to send given amount of flow units from the source into the sink in minimum possible time horizon [24, 39, 75, 79]. Thus, MDF problem and QF problem behave conversely to each other with respect to their objectives. Evacuation situation at which given number of evacuees is to shift to the *safety* as quickly as possible can be modeled as a QF problem. We discuss about this problem in Subsection 3.5.2 and extend it to a special class of network with prioritized sink terminals in Section 5.5.

Another variant of MDF problem is *earliest arrival flow (EAF) problem*. Since it is usually not known when disaster will actually happen, it is desirable to organize an evacuation in such a way that as many evacuees as possible are saved. An EAF problem aims to optimize the objective of MDF problem for every time step within pre-specified time horizon. The solution procedure given in [91] for EAF problem modeled for two terminal series parallel network has been discussed in Subsection 4.4.1. We adopt the same idea to propose solutions for EAF problems extended in this thesis. For deeper insight in the problem, we refer to [7], [44], [80] and [103].

1.2 Objective of Thesis

The objective of this thesis is to enhance the network flow optimization models for evacuation planning problems incorporating the real world evacuation attributes, and search for efficient solutions for these problems. We aim to solve the network contraflow evacuation planning problems modeled on network with capacitated destinations of given priority order. In particular, our focus is in solving maximum versions of the problems, and extending their solutions to solve the problems in quickest and earliest versions. These solutions give efficient way out to evacuate people in risk zone during disasters. Thus, our interest for studying these optimization problems is to minimize the human casualties due to disasters.

Contraflow Approach

During evacuation, the major objective is to get all people out of risk zones in minimum possible time before disaster. One of the potential ways to accelerate the process of achieving this objective is to apply notion of network contraflow in evacuation plan. One obvious reason for motivation to adapt this approach is the fact that the movement towards risk zone is undesired, and these lanes remain unused during evacuation. Evacuation plans with this notion allow direction reversal of unused lanes that significantly increase capacity of routes in the direction of flip.

Introducing the notion of contraflow in network, Rebennack et al. [89] studied MDF problem and showed that there exists analytical solution also for the problem. They gave solution with polynomial complexity for the problem investigated on both static and dynamic networks where arc reversal ability has been adapted only once at very beginning of time horizon for dynamic case. Their model is based on discrete time setting in which given evacuation time horizon is discretized in to discrete time points starting at time period zero. Moreover, structure of the network is assumed not to be changed over time. That is, number of vertices and arcs as well as arc capacities and arc transit times are fixed on the network throughout time horizon.

A couple of years before the investigation of analytical solution, dynamic network contraflow evacuation planning problem was formulated as an integer programming formulation in [68]. Two heuristics based on empirical results, Greedy heuristic which determines condition of congestion and flips highly congested arc in greedy manner and Bottleneck Relief heuristic which identifies bottleneck on the network and increases capacity by contraflow to improve maximum flow in each iteration, have also been investigated.

We study the maximum dynamic contraflow problem with continuous time setting. By applying the notion of natural transformation [40], we propose an efficient solution to this problem. Details has been presented in Subsection 4.4.2. This work has also appeared in [64].

Authors in [89] also studied the quickest version of dynamic contraflow problem and showed that this problem can be solved efficiently using the maximum dynamic contraflow solution together with solution idea of quickest flow problem [24]. Network contraflow problem with the earliest arrival feature, an important aspect for evacuation modeling, is earliest arrival contraflow problem. This problem has been studied in [31].

The contraflow problems discussed so far are not capable of modeling evacuation scenarios at which there are unequal transit time in to-and-fro direction of roads. Also, model for these problems do not allow parallel lanes of different transit time. But these are quite obvious situations that could exist due to diversity of the road network topology. Therefore, it is crucial to study network contraflow problems with not necessarily equal transit time in to-and-fro direction of arcs. With this consideration, we study some contraflow problems, namely, maximum dynamic contraflow (MDCF) problem and earliest arrival contraflow (EACF) problem over multi-network. Problems over multi-network capture the situation with parallel lanes of different transit times on roads. We propose efficient solution procedures to the problems modeled with discrete as well as continuous time settings. Details has been presented in Chapter 4. The preliminary results on these problems are also included in [12], [14], [16], [18] and [66].

Capacitated Prioritized Vertices

It is usually not known in advance how many people are at risk zone and how many people does road topology immediately after disaster (e.g., earthquakes) allow to send in to destination in specified time period. Thus, it is advisable to organize an evacuation such that as many evacuees as possible can also be sent to relatively safe places away from risk zone despite of sending them to safe destination, the sink. Modeling evacuation problem with this aspect is not obvious in the existing network flow models due to flow conservation constraint at intermediate vertices. Instead, this property is to be switched off and set weak-conservation constraints in the model so that it allows to hold flow units at intermediate vertices, also.

We study MDF problem with capacitated vertices and call it *lexicographic maximum dynamic flow (LexMDF) problem*. This problem is motivated by the situation encountered in evacuation scenarios: as many evacuees as possible are to be sent to *safety*. However, if sending evacuees to *safety* is not possible within given time frame, it is often desirable to send as many evacuees as possible to temporary shelters of given capacity which are located within the evacuation zone. In LexMDF problem, we are looking for a dynamic flow which sends as many flow units from source to sink as possible and, as a secondary objective, a maximum number of flow units to vertices other than the sink. The latter is subjected to a prioritization of the vertices (terminals) with vertex capacities, delimiting the amount of flow that can be stored at them within given time horizon. The source and the sink get sufficient vertex storage capacities. We assume that the vertices are sorted from higher to lower priority. This sorting reflects the fact that certain destinations in an evacuation process have different priority: staying in the source is of least priority, while reaching the *safety* has highest priority. The consideration of vertex capacity in the model indicates whether some vertex, other than the source, serves as a shelter, i.e. if vertex capacity is positive, then the vertex serves as a shelter.

We call the maximum flow problem with capacitated vertices over static network a *lexicographic maximum static flow (LexMSF) problem*. We propose a solution technique based on network transformation that solves LexMSF problem efficiently, whereas solution procedure proposed for LexMDF problem that is based on time-expanded networks runs only with pseudo polynomial time complexity. However, we are able to solve the latter problem modeled over a uniform path length (UPL) network in strongly polynomial time by using temporally repeated flows.

As an extension of LexMDF problem, we aim to achieve the objective of sending as many evacuees as possible from risk zone to the *safety* at each time step within specified time horizon. For this we study *lexicographic earliest arrival flow (LexEAF) problem*. We propose a polynomial time solution procedure for this problem modeled over a restricted class of network known as uniform path length two terminal series parallel (UPL-TTSP) network. Moreover, we study dynamic flow problem over the network with prioritized vertices at which we aim to minimize the time that are required to fulfill demands at these vertices in given order. We call this dynamic flow problem a *lexicographic quickest flow (LexQF) problem*. We propose a polynomial time solution procedure to this problem modeled over UPL network. We also study some contraflow evacuation planning problems–maximum contraflow problem for static as well as dynamic case and earliest arrival contraflow problem for UPL-TTSP network, with capacitated vertices. Dynamic flow problems with capacitated vertices are considered with discrete time setting and indicated that the solutions to these problems can be modified to solve the problems with continuous time setting, also.

Evacuation models with capacitated vertices and solution strategies for them are given in Chapter 5. The preliminary concepts, models and results are also included in [11], [13], [15], [17], [18], [19], [63], [65] and [66].

1.3 Rationale of Thesis

The network flow optimization models we propose in this thesis are primarily applicable in modeling the evacuation scenarios. Models and their solutions are useful in analyzing the lower bound for evacuation time or the upper bound for number of evacuees, and in finding the bottleneck of the evacuation network at where special attention can be given for building infrastructures for future evacuation planning. Despite the evacuation planning problems, network flow models, imposing necessary conditions on the model, are useful to model other transportation management problems aiming to relieve day-to-day traffic congestion, to mitigate the directionally imbalanced traffic, to reduce traveler's journey time, to reduce carbon emission and energy consumption and to increase productivity of transportation investment. Moreover, standard network flows are applicable in modeling a variety of optimization problems, including manufacturing, scheduling and routing, telecommunication, financial flows and most branches of engineering fields [2].

1.4 Outline and Assumptions

We shall proceed discussing the rest of the thesis according to the following plan. In Chapter 2, disaster management techniques found in literature are introduced in brief and importance of efficient evacuation planning during disaster is highlighted. Chapter 3 includes notations, definitions and tools; and basic network flow problems with their solutions that play vital role for developing algorithms and prove their correctness as well as optimality for the problems introduced in consecutive chapters. Evacuation planning problems based on contraflow approach are discussed in Chapter 4. In particular, a continuous time maximum dynamic flow evacuation planning problem with efficient solution has been given. Also, a new contraflow based evacuation model has been introduced and efficient solutions for various problems over multi-network have been proposed. Again, in Chapter 5, a new flow model in which flow may not be conserved at intermediate vertices has been introduced. This chapter deals with evacuation planning problems modeled on the network with prioritized terminals of given capacities. We have given polynomial and pseudo-polynomial time algorithms, respectively, for static and dynamic version of lexicographic maximum flow problem with capacitated vertices for general network. Polynomial solutions have also been proposed for dynamic flow problems on special network known as UPL network and UPL-TTSP network. Also, Chapter 5 includes models and solutions of evacuation planning problem, in which we combine contraflow approach in evacuation model with capacitated vertices. Chapter 6 concludes the thesis with future research directions in the topic.

It has been tried to make a self-complete thesis in the sense that models and their solution techniques (algorithms) that we mention are explained in detail. Examples and figures support the explanation in most of the cases. Also, algorithmic proofs for correctness and optimality of the proposed solution are given. However, it is advantageous for the reader to become familiar with the notion of Graph Theory and classical network flow theory; flow models, concept of cuts, analysis of the performance of algorithms, etc. For a broader insight into network flow problems, we refer to seminal books [2, 43] and survey articles [6, 73]. Concept of Measure Theory is helpful in understanding the continuous version of dynamic network flow problems.

Throughout the thesis discussion we make the following assumptions unless stated otherwise. The flow units travel in a uniform speed throughout time horizon considered at each moment. That is, transit time that a flow unit takes to travel an arc is invariant of congestion on the arc as well as time. All network input parameters are considered integral. Flow may or may not be conserved at intermediate vertices, and intermediate vertices may or may not have storage capacities depending on the problem and will be stated explicitly wherever required. Capacities on arcs and vertices and transit times are constant. Time parameter is taken in discrete as well as continuous manner and are stated clearly wherever necessary. Transit time parameter on arcs behave symmetrically during contraflow process. We consider variety of networks; two terminal general networks, two terminal series parallel (TTSP) networks, uniform path length (UPL) networks, UPL-TTSP networks and multi-networks to model the evacuation problems of different types. We occasionally call 'ordinary network', if it is not a multi-network. However, each of these networks has a single source and a single sink.

CHAPTER 2

DISASTER MANAGEMENT AND EVACUATION PLANNING PROBLEM

2.1 Introduction

Disaster, an impact of hazard, is a sudden and serious disruption of functioning of community or society which exceeds ability of affected community or society to remain in normal life. Natural as well as human caused hazards have been threatening lives and property and causing injuries, diseases and environmental damages. Hyogo Framework for Action [105] defines a hazard as

> *"a potentially damaging physical event, phenomenon or human activity that may cause the loss of life or injury, property damage, social and economic disruption or environmental degradation".*

Disasters seem to be major problem worldwide and a serious threat to sustainable development since it can also cause social and worldwide economic disruption [100]. During the past decade, these disasters have become more frequent and devastating. Thus, disaster management model is crucial and has become topic of interest to diverse sectors such as authorities, stakeholders, researchers and academia. The model simplifies the complex disaster scenario, compares real situation with theoretical models, quantifies events and establishes a common base of understanding for all stakeholders [62]. Disaster management is the systematic process of handling the resources and responsibilities for dealing with all humanitarian aspects of emergencies due to hazards, in particular prevention, preparedness, response and recovery (PPRR) in order to lessen the impact of disasters. The PPRR model is a comprehensive approach that attempts to consider all necessary steps required throughout the process of disaster management. However, various other models for disaster management classified as logical models, integrated models, cause models, combinatorial models and miscellaneous models have been developed to address particular disaster scenarios [82, 109]. For a primary response, an evacuation planning problem gives an efficient way-out on existing road network that attempts to shift all evacuees from risk zone to safer zone in minimum time with minimum casualty during the disasters.

A mega earthquake (7.8 Richter Scale) struck central Nepal on 25 April 2015, a few weeks before the proposal of this thesis was drafted, killing more than 8,000 people, making more than 22000 people severely injured and destroying a quarter of a million homes [111]. About a hundred thousand people died during or as a result of the earthquake that hit Haiti on January 2010 [72]. 2004 Indian Ocean earthquake and tsunami, known as the deadliest natural disaster, killed about two hundred twenty seven thousand people from more than a dozen of countries [51, 97]. These are only a few examples among hundreds of such tragic hazards caused by nature (e.g., floods, earthquakes, landslides, windstorms) in a few decades. Besides, hazards with similar tragic impact may be biological (e.g., communicable disease outbreaks, threat of virus attack), industrial accidents (e.g., chemical spills, fires, explosions), transportation accidents, bomb threat, etc; most of which are due to human interventions. Until the last day of April 2020, while summing up this thesis, the pandemic Covid-19 killed more than three hundred forty two thousand people across the world, and this number is increasing [112]. Sometimes one disaster triggers another disaster, such as an earthquake mostly triggers landslides and floods. Moreover, in some cases, the cause of a disaster will be complex; it will be human-made events exacerbating natural events, or vice-versa.

Cities are experiencing the congestion even without the threat of disasters day by day. The United Nations [104] estimates that more than two-thirds of the world's population will live in cities by 2050. About 50% of the world's urban dwellers reside in settlements with fewer than 500,000 inhabitants, while around one out of eight lives in 33 mega-cities with more than 10 million inhabitants. By 2030, the world is projected to have 43 mega-cities, most of them in developing regions. Rapid urbanization and population growth are also driving the increase in disaster risks. Hence, traffic will increase exponentially once people are aware of the risk and the emergency response becomes more complicated. Besides the rapid urbanization causing increased population density, environmental degradation, lack of awareness or preparation for common disasters, poverty, wars and civil conflicts and socio/political issues can heighten the effects of a disaster. The biggest problem with the disaster is suddenness and swiftness with which they occur.

Disaster can occur suddenly, but its recovery and reconstruction may take years and even decades. A proper disaster management could significantly mitigate possible devastating impact of hazards in such unwanted situations and help in building disaster resilience community. UNISDR [106] has defined the term 'resilience' as

> *"the ability of a system, community or society exposed to hazards to resist, absorb, accommodate to and recover from the effects of a hazard in a timely and efficient manner, including through the preservation and restoration of its essential basic structures and functions".*

2.2 Components of Disaster Management

Hazards cannot be prevented totally, but disasters can be reduced by proper planning and precautions. They are generally seen as extreme events in their scale or impact, requiring some form of external assistance. Disaster management aims to achieve early evacuation of victims and rapid and durable recovery from the impact of hazards. A poorly planned disaster relief activities can have a significant negative impact on disaster victims.

An integrated action plan is required for effective disaster management. Such plans are multi-layered (typically, four phases) and are planned to address the issues caused by hazards in order of risk incidents and hence tells about activities that should take place before, during or after an incident. That is, we need to be careful in how one can prevent, prepare for, respond to, and recover from the certain disastrous events. All these measures are crucial during emergency with respect to their own characteristics, and thus, should be acted in such a way that the hazard cannot take the form of disaster. As the successor instrument to the Hyogo Framework for Action (HFA) 2005-2015, the Sendai Framework (SF) 2015-2030 [107] has set objective for substantial reduction of disaster risk and its consequences. SF has also set the following goal to achieve the objective:

> *"Prevent new and reduce existing disaster risk through the implementation of integrated and inclusive economic, structural, legal, social, health, cultural, educational, environmental, technological, political and institutional measures that*

prevent and reduce hazard exposure and vulnerability to disaster, increase preparedness for response and recovery, and thus strengthen resilience."

We explain briefly the four fundamental phases (PPRR) of disaster management in the following, however, its measures vary from one disaster to another.

Prevention

Proactive measures designed and acted at the disaster prone areas and for the people of these areas fall in the category of disaster prevention. It aims to provide a permanent protection from or to get rid of the impact of disasters, to ensure that human activities or natural phenomena do not lead to disaster or emergency or to reduce the severity of the lives and loss of properties caused by the disaster. Hazard identification is the primary issues in prevention planning for appropriate strategies to carry out. Not all disasters, particularly natural disasters, can be prevented, but their scale or severity can be substantially reduced or mitigated by taking necessary precautions and managing them. For example, some severe weather events can be forecasted hours before they occur, providing significant time for precautions. Prevention includes long term activities which reduce the effects of unavoidable disasters. Such activities are expected to be cost effective. Preventive tools can be categorized as structural that includes the reinforcement or replacement of inadequate buildings or structures, construction of dams and fences at the disaster prone regions, and nonstructural that includes legislation, by-laws, development and application of building codes and public awareness. Moreover, preventing habitation in disaster prone regions or isolating them from potential sources of disasters from the habitation could reduce the effect of disaster. Some intentional human caused disasters (terrorist attacks for example) can be prevented or mitigated their impact by the regular monitoring and detection from authorities.

Preparedness

Disaster preparedness refers to the tools considered to be developed before disasters so that their effects could be reduced. That is, it aims to predict and prevent the disaster where possible, mitigate their impact on vulnerable people; and respond to and effectively cope with their consequences. It is mandatory to develop a policy that states a detailed mechanisms and structures for effective and practical management of disasters. A well developed preparedness scheme is expected to address not only immediate response, but long-term recovery and reconstruction also. An evacuee may not have sufficient experience of disasters due to the unpredictability and suddenness of the occurrence of a disaster. Thus, real evacuation, being a unique transportation activity (dissimilar to their ordinary travel behavior) under emergency situations, the frequent rehearsal, mock-up drills, and simulation exercises of artificial evacuation scenarios to the resident of disaster prone region are important. Besides, activities such as preparing good evacuation plans in advance, forecasting of disaster and managing early warning system wherever possible, preparing effective rescue plan and identifying the feasible open spaces with basic necessities for rehabilitation are expected in disaster preparedness phase. Public awareness programs shall be carried out through the effective and maximum use of media, interactions and consultations, education and other informal means to sensitize the people about disasters. It is necessary to prepare hazard-risk maps of disaster prone areas and to make known to related residents and stakeholders. An adequate stocks of rescue and relief materials shall also be stored at appropriate locations. Most importantly, sufficient volunteers are to be trained for implementation of emergency disaster management plan. The primary respondents of a disaster are those people living in local community. Thus, it is expected to focus on community-based disaster preparedness, which assists communities to reduce their vulnerability to disasters and strengthen their capacities to resist them.

For long term preparedness, disaster management shall be recognized as a national and local level priority, and shall be taken as a continued and sustainable activity. The emphasis shall be on promotion of culture of safety, strengthening an optimum utilization of local skills, resources, capacities and technologies to lessen the impact of future disasters. It is required to strengthen and extend facilities for research and development in the fields of seismology, hydrology and meteorology and remote sensing for disaster preparedness and mitigation, including early warning system. Educational institutes shall be encouraged to include disaster related curriculum in formal education.

Response

All the actions that are taken from initial minutes of disaster to the time of complete recovery from it fall upon response phase. Systematic and coordinated multi-agency responses are required to reduce impact of disaster and its long-term results. Vulnerability assessment is the primary issue in response plan since an understanding of what can happen due to disaster enables to determine resource requirements and to develop plans and procedures for effective response. Response action plans taken immediately after an emergency are critical and shall be immediately acted after the preliminary loss and damage assessment. When an emergency occurs, the first priority is always safety of lives. Therefore, emergency evacuation is crucial during response phase. It is required to identify new open spaces for shelter if pre-located open spaces became inappropriate right after the disaster. In addition to the efficient evacuation, there could be different activities that help in reducing the impact of disasters. For example, the First Aid and CPR (cardio-pulmonary resuscitation) by trained personnel can save lives, use of fire extinguishers can extinguish a small fire, and supervision of building utilities and systems can minimize damage to a building and help prevent environmental damage. Some other activities which are crucial in this phase are to prevent people from disease and disability, provide temporary shelter with basic necessities such as food, safe drinking water, proper sanitary facilities and emergency health care.

Recovery

Once the initial crisis due to disaster is over and basic emergency needs have been fulfilled, it is expected to run short term and/or long term recovery activities to turn affected people back to normalcy. For this, a disaster recovery plan is to be drafted which could be multifaceted and complex. The plan contains a detailed damage assessment of disaster and instructions on how to respond to them. The ideal disaster recovery process recognizes possibilities of situation, and manages necessary activities so that impacts of disaster could be minimized. Rebuilding and/or retrofitting infrastructures with standard codes, replacing property and resuming employment recovers the impact gradually. Recovery activities also include security and health care facilities till the impacted people are fully recovered and rehabilitated. Psychological counseling for affected people is necessary to get rid of fear and torture of impact of disaster. Tortures can also be pacified by providing suitable grants and/or soft loan packages, or introducing income generating activities to affected people. Environment recovery also plays important role in overall recovery process since without it the victims cannot be cured psychologically.

Thus, in any type of hazards, it is almost impossible to avoid disasters entirely. It should be noted that prevention strategies are generally less costly in social and economic terms than emergency responses. Therefore, stakeholders should pay more attention on preventive measures. During all phases of disaster management, special priority shall be given but not limited to the population with special needs such as children, women, senior citizens, differently able and underprivileged groups. Both preparedness and response phases of the disaster management include activity of rescuing people, shortly before and/or after occurrence of hazard, from risk zone to safer zone. However, during earthquake, for example, as it is not sure when and where it occurs, it may not be possible to rescue people in advance. In such situations, only an emergency response is to be made from the authority to lessen the impact of hazard. The primary objective of an emergency response is to apply a proper evacuation plan that could mitigate possible devastating impact of a disaster, specially minimizing loss of lives. Since recovery activities has long-lasting effects and usually takes high costs, participants in the process are from various sector and large in number. However, recovery phase provides opportunities for community development, especially in terms of creating sustainable, safer, and more resilient communities.

2.3 Evacuation Planning Problem

An urgent evacuation is the quick escape of people away from an area with threat of disaster to save lives. Evacuation, as an intuitive and practically effective emergency rescue tool, has long been used and is expected to be enhanced to protect human lives and their property from hazards. In general, the purpose of an evacuation plan is to maximize the utilization of an existing transportation system so that the safest tour and most efficient evacuation time of all expected evacuees of a building, city, or region could be ensured.

Evacuation is inherently a transportation process occurring in an existing road network. However, depending up on characteristics of the models and their solution strategies, it has been categorized in two major groups: optimization-based, known as macroscopic models and simulation-based, known as microscopic models.

An optimization-based model is indifferent of individual's decisions and has capability of identifying optimal scenarios in systematic and self-driven manner to search for optimal evacuation plans, mainly based on solutions using exact methods or heuristics. Optimization-based evacuation planning problems can be modeled in a mathematical programming form and whole system is governed by fixed set of constraints. These models provide good lower bound for evacuation time, as they aim at finding a system optimum by assuming the cooperative behavior of group of evacuees. Literature is flourished with variety of optimization based evacuation problems with exact solutions. We refer to survey articles [6, 73, 94] for the literature. There exist heuristic (considerably good) solutions also for some very difficult (e.g., NP-hard) optimization evacuation problems. See, e.g., [20], [35], [60], [67], [68], [98], [99], [108], [110], etc. for deeper insight into this class of solutions. The biggest limitation of a heuristic is that it cannot guarantee optimality in general despite of investing several iterations. Another dimension for dealing the evacuation planning problem is to apply the means of evolutionary computations. Evolutionary algorithms are also heuristic-based approach for solving problems that are not polynomially solvable. Evolutionary computations in the form of Genetic Algorithms (GAs) and Estimation of Distribution Algorithms (EDAs) are used to manage exits in order to optimize overall evacuation time [46]. GAs are popular robust optimizer that simulate the processes in natural evolution system and show better performance in searching optimal or near to optimal solutions to the search problems [37, 46]. For deeper insight into implementation of GA, its variants (Evolution Strategies and Evolutionary Programming), and other means of evolutionary techniques (Particle Swarm Optimization, Ant Colony Optimization, etc.) in emergency management and evacuation planning, we refer to the survey article [109].

In simulation-based model, an individual evacuee's movement is emphasized rather than system optimization. These models are capable of showing evacuation process in a detailed, disaggregated and distributed manner. These models lead to solutions with more accuracy, but usually suffer from high computation times. It is important to keep in mind that a well-defined and tractable plan is one of the prerequisites for successful implementation of a large-scale evacuation. One may encounter with some mesoscopic models as well in literature which is the integrated model of macroscopic model and microscopic model.

An evacuation setting can be modeled and optimized using network flows. A network is a directed graph consisting of the set of arcs and vertices. Vertices represent the junction on the city evacuation, rooms in a building evacuation, seats in board evacuation and so on. Arcs connect locations represented by vertices. They represent the road segments and ways from one room to another in a building, etc. Number of lanes on the road can be taken as the capacity of the arc and the time taken to travel from one vertex to its adjacent vertex is its transit time. Risk zone where disaster has occurred or going to occur soon is taken as the source and the safe place where the people are to be evacuated is assumed to be the sink. The source contains evacuees, and the sink waits them for shelter with enough space. The movement of evacuees on the road is considered as the flow on the network.

Evacuation network flow problems could be of different flavors depending upon the evacuation settings: available resources, sensitivity of disaster, needs of evacuees, etc. Variety of models and solution techniques for evacuation tasks, depending upon problem size, behavioral and organizational situations, modes of transportation and traffic capacity, time dependency, origin-destination assignment, evacuation objectives, contraflow, etc., do exist in literature [30, 83, 73]. Static version of network flow models is not capable to capture real world evacuation scenarios in most of the cases. Objectives of the model differs one problem to another depending upon evacuation settings. One may be interested in shifting as many people as possible in a fixed time period whereas another may be in shifting a given number of people in minimum time period or shifting them as earlier as possible. Achieving these objectives in an efficient way is not always an easy nut to crack. The difficulty depends upon network architecture, road attributes (e.g., congestion) and types of parameters (e.g., capacities, transit times) we consider for modeling the problem. It also depends upon the type of solution we desire.

2.4 Stakeholders

The Sendai Framework (SF) 2015-2030 [107] states that disaster risk reduction strategies require that the responsibilities be shared by central governments, relevant national authorities and other stakeholders with respect to national circumstances and systems of governance. Stakeholders for efficient disaster management include all levels of government, business community, political leadership, community activists, academia and individuals. Their roles shall also be clearly defined through legal provisions. Public as well as private investment in disaster management through structural and non-structural measures are essential to enhance economic, social, health and cultural resilience of people and communities by means of innovation, growth and job creation. Academia, such as universities, research centers and the people related to these, play major role in planning and implementation of necessary efforts since these require a scientific base whereby the proper information and the uncertainties of disaster impact could be fully understood before the preparedness plan can be translated into policy and regulatory measures. Universities and research centers are often, but not always, sources of pertinent information and are developers of analytical and critical methods and tools. If these potential assets are combined with strong management skills typically found in private sector, then academia can make major and lasting contributions.

Development of mathematical models for evacuation planning reflecting real world evacuation scenarios with their efficient solutions, rehearsal of plans and their implementation are crucial for saving lives during disasters. For effective implementation, it is important to update data-set of the number of evacuees at disastrous zone and their status, open feasible spaces for shelters with basic necessities and existing usable road network. This thesis focuses on developing evacuation models based on network flow optimization problems and discusses solution techniques to these problems.

CHAPTER 3

BASIC NETWORK FLOW PROBLEMS

3.1 Introduction

Network flow theory is one of the most flourished fields of optimization, and has important applications to quite different branches of science and technology, besides obvious applications to transportation and scheduling [59]. In network flow problems, the objective is mainly to move some entity from one point to another through a network using the available transmission facilities efficiently. Therefore, these are also applicable in modeling the evacuation planning problems. In this chapter, starting with introduction of evacuation network, we discuss some fundamental network flow problems extracted from literature. We also discuss solution techniques to some of these problems which become building blocks for constructing the solutions to the evacuation problems developed in Chapters 4 and 5.

3.2 Basic Denotations and Definitions

Mathematical optimization based evacuation planing problems are modeled as network flow problems. A network $\mathcal{N}$ is defined to be a directed graph $G = (V, A)$ consisting of a set of vertices V and a set of directed arcs $A = V \times V$. Two subsets: set of sources $\mathcal{S}^+$ and set of sinks $\mathcal{S}^-$, of vertex set V are specified. The remaining vertices $v \in V \backslash (\mathcal{S}^+ \cup \mathcal{S}^-)$ are called intermediate vertices. We may refer to an arc $a \in A$ by specifying its start and end vertex, i.e. $a = (v, w)$ with $v \neq w$ and $v, w \in V$ as tail and head, respectively. While modeling the evacuation scenario, arcs and vertices on network $\mathcal{N}$ refer to the lanes of road and their intersections, respectively. Both, set of vertices and set of arcs, are assumed to be finite and we set $n := |V|$ and $m := |A|$. The movement of evacuees on lanes is assumed to be flow on the arcs. Along with network $\mathcal{N}$, we assume lower and upper arc capacity function $l, u : A \to \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ to be given, which bounds the number of flow units on each arc at each time step from below and from above. Most of the time, we set $l(a) = 0$ for all $a \in A$. Similarly, the vertex capacity function is introduced: *vertex capacity function* $k : V \to \mathbb{N}_0$ that delimits total number of flow units which may be held at vertices. Moreover, the *arc cost function* $c : A \to \mathbb{N}_0$ specifies the cost c for a flow unit to travel an arc $a \in A$. Further, transit time function $\tau : A \to \mathbb{N}_0$ ($\tau : A \to \mathbb{R}_{\geq 0}$ for continuous case) specifies time needed by a flow unit to traverse an arc. The transit time is considered to be antisymmetric. That is, for $(v, w) \in A$, $\tau(v, w) = -\tau(w, v)$. We sometimes refer to transit times as *length* and *cost* by stating wherever necessary. An evacuation process requires a certain time horizon $T \in \mathbb{N}$ to be given that plays a crucial role in evacuation problems. We consider the time horizon T finite[1] throughout the thesis. Summing it up, a typical dynamic evacuation network is denoted by $\mathcal{N} = (G, u(a), c(a), k(v), \tau(a), T)$. If one wishes to refer to a static network without transit times, it can just be written as $\mathcal{N} = (G, u(a), c(a), k(v))$.

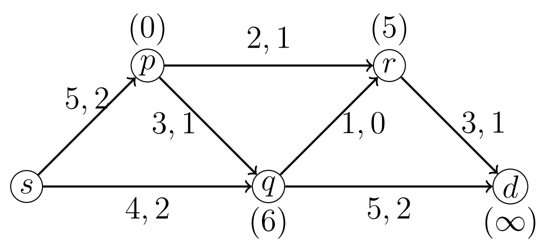


Figure 3.1: A network $\mathcal{N}$ with arc capacity and transit times next to each arc and the vertex capacity inside the parenthesis near by each vertex except the source s.

A network could have parallel arcs of different transit times. We encounter with such networks, known as multi-networks, while extending the contraflow models in Chapters 4 and 5. However, in most of the cases we consider an ordinary evacuation networks. Also, we consider the network with only a single source and a single sink in this thesis. A particular example of evacuation network with a single source single sink has been depicted in Figure 3.1. A multi-network has been depicted in Figure 4.2.

A $v_0 - v_k$ path (known as chain or route also) in a network $\mathcal{N}$, denoted by γ, is an ordered set of directed arcs $((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k))$ such that each vertex $v_i \in V$ is distinct.

[1] Flow models with infinite time horizon T also appear in a number of practical problems where activities are initiated endlessly but optimization focuses on given periods [73].

The transit time of a path γ is the sum of all the transit times of the arcs contained in it and is denoted as $\tau(\gamma)$. And, its capacity is the minimum capacity among the capacities of the arcs in it. A path γ, containing the arc (v_k, v_0) also, for $k > 1$, turns out to be a directed cycle on $\mathcal{N}$.

We also consider a special class of networks, known as two terminal series parallel (TTSP) networks as depicted in Figure 3.2, to describe some network flow problems in Chapters 4 and 5. A TTSP network $\mathcal{N}$ is a directed graph with a single source s and a single sink d which has a single arc (s, d) together with source s and sink d or is obtained from two series parallel networks $\mathcal{N}_1$ and $\mathcal{N}_2$ by one of the following operations:

(i) (Parallel Composition): Merge source vertices s_1 of $\mathcal{N}_1$ and s_2 of $\mathcal{N}_2$ to get source s of new network $\mathcal{N}$ and merge sink vertices d_1 of $\mathcal{N}_1$ and d_2 of $\mathcal{N}_2$ to get sink d of new network $\mathcal{N}$.

(ii) (Series Composition): Merge the sink vertex d_1 of $\mathcal{N}_1$ with the source vertex s_2 of $\mathcal{N}_2$.

Figure 3.2: TTSP network $\mathcal{N}_1$ with source s_1 and sink d_1 (left) and TTSP network $\mathcal{N}_2$ with source s_2 and sink d_2 (right).

Figure 3.3 illustrates the examples of TTSP networks with the construction process of network from series as well as parallel composition of the TTSP networks depicted in Figure 3.2. A network depicted in Figure 3.1 is not a TTSP network.

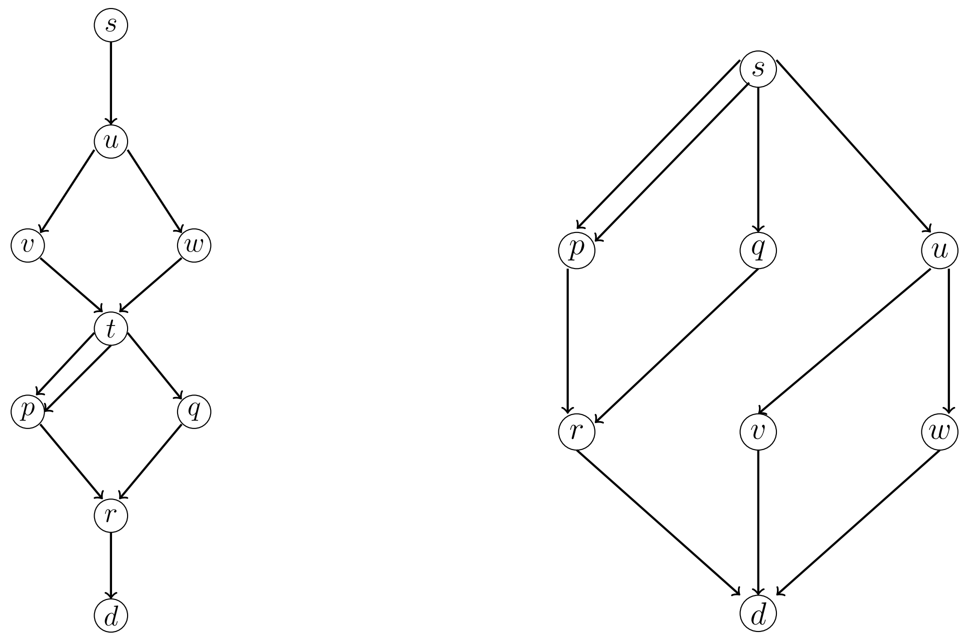


Figure 3.3: TTSP networks obtained from TTSP networks $\mathcal{N}_1$ and $\mathcal{N}_2$ depicted in Figure 3.2 using series composition (left) and parallel composition (right).

3.3 Network Flow Models

Network flow models, based on flow conservation at intermediate vertices, may be broadly classified into two categories: standard models and generalized models. The former model requires flow conservation on all arcs, that is, the amount of flow entering an arc is equal to the amount of flow leaving it. The latter model is a generalization of standard flow models in which each arc of underlying network, known as lossy network, has associated a gain or a loss factor. Here, we are concentrated only on the flow models of former category. However, we introduce flow model at which flow may not be conserved at intermediate vertices on network in Chapter 5. Study of evacuation planning problems modeled with the violation of flow conservation constraints for vertex is a substantial part of this thesis. Until we reach Chapter 5, we consider flow models with flow conservation constraints at intermediate vertices.

Based on the models and different methodologies of their solutions, network flow models could be categorized into three classes: static, dynamic, and uncertain. A good overview of static network flow models can be found in [2]. Dynamic network flow model was first introduced by Ford and Fulkerson [42] and has been developed extensively, adopting discrete as well as continuous time, in literatures, e.g. [4], [5], [7], [8], [22], [24], [39], [40], [53], [55], [56], [71], [75], [79], [89], etc. We also refer to the survey articles [6], [30] and [73] for further insights in various dynamic network flow problems and their solutions. Problems based on uncertain network flow models are usually studied by using the probability theory in [3], [81], etc. However, we consider the problem of former two types only.

3.3.1 Static flow model

Given a static network $\mathcal{N} = (V, A, u(a))$ with arc capacity function $u(a)$, non-negative flow variables $f : A \to \mathbb{N}_0$ specify the flow over arc $a \in A$ on network $\mathcal{N}$. More precisely, the number $f(a)$ represents flow units traveling along arc $a = (v, w)$ from vertex v to vertex w. The number of flow units entering arc a, for all $a \in A$, is required to be bounded by its capacity $u(a)$, i.e.,

$$0 \leq f(a) \leq u(a) \;\; \forall \;\; a \in A. \tag{3.1}$$

The total flow entering a vertex $v \in V \backslash \{s, d\}$ has to be equal to the total flow exiting out of it, i.e.,

$$\sum_{a \in \delta^-(v)} f(a) - \sum_{a \in \delta^+(v)} f(a) = 0 \;\; \forall \;\; v \in V \backslash \{s, d\}. \tag{3.2}$$

where $\delta^-(v) := \{a \in A : a = (w, v)$ for some vertex $w \in V\}$ and $\delta^+(v) := \{a \in A : a = (v, w)$ for some vertex $w \in V\}$ denote the set of arcs entering and leaving vertex $v \in V$, respectively.

The constraints given by 3.1 are known as capacity constraints and any static flow, say f_s, on $\mathcal{N}$ that fulfills these constraints is called a feasible static flow on $\mathcal{N}$. The constraints given by 3.2 are known as flow conservation constraints for vertices. A flow that satisfies the flow conservation constraints in all vertices $v \in V$ is called a circulation.

The flow function $f : A \to \mathbb{N}_0$ defines a flow on arcs. The overall goal being to optimize an evacuation problem, it is advantageous to determine paths from s to d on network along which people can leave disastrous site to become safe. A solution to any network flow problem, formulated in terms of arc flows, can be decomposed into a set of flows on paths and cycles. Moreover, the Flow Decomposition Theorem shows that the flow model based on arc flows and the flow model based on path and cycle flows are equivalent [2]. It also suggests that for any feasible arc flow $f(a)$, there exists flow decomposition $(\Gamma, f(\gamma))$ where Γ is a set of at most $m+n$ many $s-d$ path flows and circulations on cycles on $\mathcal{N}$, and $f(\gamma) \geq 0$ for each $\gamma \in \Gamma$ is a flow value corresponding to path γ. Thus, for every directed $s-d$ path or cycle γ, it specifies a flow value $f(\gamma)$ such that every arc a has at most $u(a)$ units of flow passing through it. For these flow values on paths and cycles, the following property has to hold:

$$f(a) = \sum_{\gamma \in \Gamma : a \in \gamma} f(\gamma) \qquad \forall a \in A.$$

The path decomposition $(\Gamma, f(\gamma))$ is known as *standard path decomposition* of arc flow $f(a)$ if all of its path flows use arcs in the same direction as $f(a)$ does. The path decomposition is carried out by using the algorithm for finding a maximum $s-d$ flow backwards (cf. [2]). Here, the network consisting of arcs $a \in A$ having a positive flow $f(a)$ only is considered. In this restricted network, say $\mathcal{N}^*$, the capacity $u^*(a) := f(a)$ is set and the algorithm to find a maximum $s-d$ flow is applied. In each iteration, at least one arc is discarded (residual capacity decreased to zero) from corresponding network $\mathcal{N}^*$ and therefore at most m many paths can be found. The resulting set of paths together with the corresponding minimal residual capacity builds a path decomposition. Flow that is still on cycles can be ignored (deleted from the network) since it does not increase the amount of flow sent from source s to sink d. The notion of path decomposition is applicable in finding temporally repeated flows while solving dynamic version of network flow problems.

3.3.2 Dynamic flow model

Evacuation plans reflecting nearer to real situation can be modeled with the help of the dynamic network flow model. In contrast to the static network flow model, it is based on network with time parameter on it where the transmission of flow is not instantaneous, rather it progresses as time progresses. Moreover, in such networks, flow units can be delayed at vertices, and network parameters, e.g., arc capacities, can change in time. The phrase *flows over time* is also used in literature for dynamic flows. For T being the pre-specified finite time horizon and $\tau(a)$ being the time needed by a flow unit to traverse an arc a for all $a \in A$, the flow rate $f(a,t)$ on $\mathcal{N} = (V, A, u(a), \tau(a), T)$ defined as $f : A \times [0, T) \to \mathbb{N}_0$ is the amount of flow entering the particular arc per time unit. Further, flow that enters arc a at time t, reaches the head of arc a at time $t + \tau(a)$. The flow rate for each arc $a \in A$ should satisfy capacity constraint at each time point t within T, i.e.,

$$0 \leq f(a,t) \leq u(a) \;\; \forall \;\; a \in A \;\; \text{and} \;\; \forall \;\; t \in [0,T). \tag{3.3}$$

The flow function satisfies the flow conservation constraints: For each vertex $v \in V\backslash\{s,d\}$, total flow units that enters into v must be equal to the total flow units exiting out of it within T, i.e.,

$$\sum_{a \in \delta^-(v)} \int_{\xi=0}^{T-\tau(a)} f(a,\xi) - \sum_{a \in \delta^+(v)} \int_{\xi=0}^{T} f(a,\xi) = 0 \;\; \forall \;\; v \in V\backslash\{s,d\}. \tag{3.4}$$

Any continuous dynamic flow f_c on network $\mathcal{N}$ that satisfies the capacity constraints (3.3) for all arcs at every time point $t \in [0,T)$ is a feasible continuous dynamic flow.

Time component being continuous in nature, the evacuation problems modeled with continuous time setting better reflect the real world evacuation tasks. However, problems modeled on discrete-time setting are easier to solve. In a discrete dynamic flow model the transit time τ are considered as integral value and the time horizon T is treated in a discrete manner and send packets of flow units at time steps (layers) in $\mathcal{T} := \{0,1,\ldots,T\}$ into the arcs instead of sending flow at continuous flow rates. In this model, we are looking for a flow function $f(a,t)$ defined as $f : A \times \mathcal{T} \to \mathbb{N}_0$ that determines the amount of flow sent into arc $a = (v,w)$ at time step $t \in \mathcal{T}$. Flow units sent into an arc a at time t should be bounded by its capacity $u(a)$ at each time step $t \in \mathcal{T}$ and totally reach the head of that arc at time $t+\tau(a)$. Flow conservation at each intermediate vertex can be adapted directly to this model also. Here, instead of integrating the flow units over time in constraints 3.4, it is required to sum over the discretized time steps $\mathcal{T}$. That is, the flow conservation constraint at vertex $v, \; \forall \; v \in V\backslash\{s,d\}$ in the discrete flow model is

$$\sum_{a \in \delta^-(v)} \sum_{\xi=0}^{T-\tau(a)} f(a,\xi) = \sum_{a \in \delta^+(v)} \sum_{\xi=0}^{T} f(a,\xi) = 0 \;\; \forall \;\; v \in V\backslash\{s,d\}. \tag{3.5}$$

Additionally, if one wishes to allow holding of flow units at intermediate vertices of given capacities up to some intermediate time points within the time horizon T, the flow function should satisfy the following constraints:

$$\sum_{a \in \delta^-(v)} \sum_{\xi=0}^{t-\tau(a)} f(a,\xi) = \sum_{a \in \delta^+(v)} \sum_{\xi=0}^{t} f(a,\xi) \geq 0 \;\; \forall \;\; v \in V\backslash\{s,d\}, \;\; t \in \mathcal{T}. \tag{3.6}$$

The storage of flow units at intermediate vertices up to time horizon T has been considered in flow models in Chapter 5.

3.4 Static Flow Problems

In this section we discuss about some fundamental network flow problems with their solution ideas. Basically, these problems and related results stand as the foundation for solution structure, their correctness proofs and optimality proofs of evacuation planning problems covered in Chapters 4 and 5.

3.4.1 Maximum static $s-d$ flow problem

Despite of wide range of applications of its own, static network flow problems are the basis of dynamic network flow problems. Literature has been flourished with this class of problems with variety of models and their efficient solution methods since the time of study of problem in [41]. Consider a two terminal network $\mathcal{N} = (V, A, u(a), s, d)$ where s and d denote the single source and the single sink, respectively. Then the objective of a *maximum s-d flow problem* on $\mathcal{N}$ is to maximize a net feasible static flow value $\mathbf{f}_s$ induced by the static flow f_s on $\mathcal{N}$, from the source s to the sink d, given by

$$\mathbf{f}_s = \sum_{a \in \delta^-(d)} f(a) - \sum_{a \in \delta^+(d)} f(a). \tag{3.7}$$

This problem is also called *maximum static flow (MSF) problem* on single-source-single-sink network $\mathcal{N}$. MSF problem with multiple sources and multiple sinks can also be similarly defined [73]. Evacuation planning problems can be modeled as a maximum static flow problem, if their objectives can be accomplished in a single wave. For evacuation plans, the supply at source vertex can be used to model the number of evacuees in respective risk zone. Such evacuation planning problem can be solved efficiently by applying either of the methods found in literature for solving maximum static flow problem.

Before discussing the solution techniques to solve an MSF problem, we illustrate notion of $s-d$ cut over network $\mathcal{N}$.

Definition 3.4.1 ***[Minimum Cuts]*** *Let $X \subsetneq V$ be a set of vertices on $\mathcal{N}$. Then we say that X is an $s-d$ cut over $\mathcal{N}$, if $s \in X$ and $d \in V \backslash X$. The capacity $u(X)$ of an $s-d$ cut X is defined as follows:*

$$u(X) := \sum_{a \in \delta^+(X)} u(a)$$

where $\delta^+(X)$ denotes set of directed arcs from vertices in X to vertices in $V \backslash X$. If the capacity $u(X)$ is minimal over all sets $X \subsetneq V$, then we call X a minimum $s-d$ cut.

The following theorem, known as *max-flow min-cut theorem*, shows a fundamental relation between value of a maximum $s-d$ flow and capacity of a minimum $s-d$ cut. As a consequence of this theorem, every minimum $s-d$ cut algorithm may be employed to solve the maximum flow problem, and vice versa.

Theorem 3.4.1 *[41] For any static network, the maximal flow value from s to d is equal to the minimum cut capacity of all cuts separating s and d.*

Definition 3.4.2 ***[Residual Network]*** *Consider a network $\mathcal{N} = (V, A)$ with actual flow f on it. Then a residual network $\mathcal{N}_f = (V, A_f)$ of network $\mathcal{N}$, with respect to flow f, is defined as follows. If an arc $a = (v, w) \in A$, and there exist a positive flow $f(v, w)$ along arc (v, w) such that $0 < f(v, w) < u(v, w)$, then A_f contains arcs (v, w) and (w, v) with residual capacities $u_r(v, w) = u(v, w) - f(v, w)$ and $u_r(w, v) = f(v, w)$, respectively. The flow $f(a)$ for the former and the latter cases are said to be forward flow and backward flow, respectively.*

Here, we briefly describe augmenting paths based algorithm given in [41] to solve maximum static flow problem. The algorithm starts with zero flow f, i.e., $f(a) = 0$ for all $a \in A$. Then it searches for augmenting $s-d$ paths as long as they exist in residual networks with respect to the actual flow f. A path from s to d in residual network $\mathcal{N}_f$ with respect to flow f is called an augmenting $s-d$ path. Along such $s-d$ path, flow amounting to minimal residual capacity, say h, is augmented, i.e., we increase $f(a)$ by h, if $a = (v, w)$ is a forward arc on the path, and we decrease $f(a)$ by h, if its backward arc (w, v) is part of the path. Since h is chosen as the minimal residual capacity of arcs along this path, we guarantee that capacity constraints 3.1 and non-negativity constraints, $f(a) \geq 0$ for all $a \in A$, are preserved in each iteration. The augmented flow f is maximum flow on network $\mathcal{N}$ if and only if there is no augmenting path in residual network $\mathcal{N}_f$.

This algorithm is simple to understand and practical to implement. Unfortunately, the total number of paths found can be exponentially large for some specially designed networks. The algorithm devised in [47], known as preflow-push algorithm, avoids this drawback since it focuses on vertices rather on arcs.

Definition 3.4.3 ***[Excess]*** *Consider a network $N = (V, A)$. The excess at vertex $v \in V$ denoted by $ex_f(v)$ is the total flow coming into the vertex v minus the total flow leaving it, i.e.,*

$$ex_f(v) = \sum_{a \in \delta^-(v)} f(a) - \sum_{a \in \delta^+(v)} f(a) \;\; \forall \;\; v \in V.$$

Definition 3.4.4 ***[Preflow]*** *Consider a network $N = (V, A)$. A preflow on N is a flow that satisfies the capacity constraints given by 3.1 and requires to satisfy the following weak conservation constraint:*

$$\sum_{a \in \delta^-(v)} f(a) - \sum_{a \in \delta^+(v)} f(a) \geq 0 \;\; \forall \;\; v \in V \backslash \{s, d\}. \tag{3.8}$$

Definition 3.4.5 ***[Active Vertex]*** *Consider a network $N = (V, A)$. A vertex $v \in V$ is an active vertex if the excess flow $ex_f(v) > 0$.*

The algorithm starting with a preflow, pushes excess flow $ex_f(v)$ at the active vertex $v \in V$ closer towards the sink. If excess flow cannot reach sink, the algorithm pushes it backwards to the source, and eventually, the preflow becomes a flow and in fact the maximum flow. The algorithm allows temporary imbalance (excess) at vertices, and works always towards feasibility.

This algorithm takes two distinct, complementary actions: change of height (label) at vertex, or push as much excess as possible to a connected vertex of lower height. We will simply say that each vertex carries a label: $l : V \to \{0, 1, \ldots, n, n+1, \ldots, 2n-1\}$. Label of the source will always be n and that of the sink be 0. These two labels will remain unaffected throughout the procedure, and serve as reference values. Other vertices will be assigned an initial label that can then be altered through a relabel operation. Labeling of any vertex (other than source and sink) always obeys one condition: if (v, w) is an arc of residual network, then we have $l(v) \leq l(w) + 1$. If vertex v has label $l(v)$ such that $0 \leq l(v) < n$, then algorithm will be working to push any excess flow at v towards the sink; but if $l(v)$ is at least as large as n, then algorithm will have reached a stage at which the excess flow at v must be pushed back towards the source and removed from network.

Running time of preflow-push algorithm is $O(mn^2)$. However, it can be easily improved to $O(n^3)$, and can be implemented in $O(mnlog(n^2/m))$ time by using dynamic tree data structure [95].

The path based solution procedure in [41] has been improved in [36] by choosing the shortest paths from source to sink in residual network, and in [34] by augmenting all the shortest paths at once in each iteration in a layered sub-network of residual network. For other polynomial time algorithms for maximum static flow problem we refer to [47].

One could solve maximum flow problem with capacities of certain arcs not fixed but functions of single parameter, known as parametric maximum flow problem, within same time bound as of ordinary maximum flow problem by extending preflow push algorithm for maximum flow problem [45]. The similar extension has been obtained in [75] for the quickest flow problem which can be viewed as a parametric minimum cost flow problem with a single source and a single sink.

3.4.1.1 MSF problem on network with multiple terminals

If an evacuation scenario is modeled over a network with multiple sources and multiple sinks with requirement: supply-demand function $\mu : \mathcal{S}^+ \cup \mathcal{S}^- \to \mathbb{N}_0$ where $\mathcal{S}^+ \subset V$ and $\mathcal{S}^- \subset V$ denote set of sources and set of sinks, respectively, we will focus on *transshipment evacuation problem*. Without loss of generality, we can assume that there is supply (negative demand) for each source vertex and demand for each sink vertex. A transshipment is a flow that fulfills supplies and demands of sources and sinks, respectively. The objective of the problem is to find such a flow on the considered network, if it exists. In transshipment problems, flow conservation has to hold only for vertices $v \in V \backslash \mathcal{S}^+ \cup \mathcal{S}^-$. For vertices $v \in \mathcal{S}^+ \cup \mathcal{S}^-$, the flow function f has to satisfy the supplies and demands $\mu(v)$. That is, f satisfies the following demand constraint (also known as the mass balance constraint) at each vertex $v \in \mathcal{S}^+ \cup \mathcal{S}^-$:

$$\sum_{a \in \delta^-(v)} f(a) - \sum_{a \in \delta^+(v)} f(a) = \mu(v); \quad \forall \; v \in \mathcal{S}^+ \cup \mathcal{S}^-. \tag{3.9}$$

A transshipment problem on network $\mathcal{N}$ with multiple sources and multiple sinks can be reduced to an $s-d$ flow problem by a slight network modification. For modification, introduce an artificial super-source s^* and an artificial super-sink d^* on $\mathcal{N}$ together with arcs (s^*, s), for all $s \in \mathcal{S}^+$ such that $u(s^*, s) = \mu(s)$ and arcs (d, d^*), for all $d \in \mathcal{S}^-$ such that $u(d, d^*) = -\mu(d)$. A feasible $s^* - d^*$ flow with value $\sum_{s \in S^+} \mu(s)$ on modified network naturally induces a feasible flow on original network satisfying all supplies and demands if a maximization of the total flow through the network is desired, Ford and Fulkerson [43]. Moreover, in such case, each maximum $s - d$ flow obviously has demanded flow value. Thus, when considering static flow models, it is sufficient to restrict to networks with single source and single sink.

The concept of maximum flow on network with an ordered set of multiple sources and multiple sinks has been defined in [77], [80] and [103]. Maximum flow problem defined on such network is known as *lexicographically maximum flow problem.* The objective of the problem is to find a feasible flow on network that lexicographically maximizes the amount of flow leaving each terminal in given priority order. This is equivalent to lexicographically minimizing the flow entering the sinks in given order.

3.4.2 Minimum cost circulations

Another static network flow problem, an extension of maximum static flow problem with an added linear cost constraint on each arc, is *minimum cost flow (MCF) problem.* The objective of the problem is to find a feasible minimum-cost flow of a specified amount from a set of supply vertices to a set of demand vertices in a directed network with linear cost functions defined on the arcs. It has numerous familiar applications in modeling transportation problems: the shortest path problems, logistics scheduling problems, etc. [2]. Besides, it often arises as a subroutine of more complex optimization problems. The problem stands as a building block of the solution techniques of the dynamic network flow problems presented in this thesis. We refer to [93] for recent survey to the problem and [74] for computational analysis of some of minimum cost flow algorithms.

Consider a network $\mathcal{N} = (G, u(a), c(a), \mu(v))$ with supplies (or demands) of amount $\mu(v)$ $\forall$ $v \in V$ such that $\sum_{v \in V} \mu(v) = 0$, and for each arc $a = (v, w) \in A$, the capacity $u(a)$ and a unit cost $c(a)$. The MCF problem in its standard form requires to find feasible (satisfying capacity constraint 3.1) flow of value $\mu(v)$ on $\mathcal{N}$, if exists, that minimizes the total cost $C(\mu(v))$

given by

$$C(\mu(v)) := \sum_{(v,w)\in A} c(v,w)f(v,w) \tag{3.10}$$

subject to

$$\sum_{w:(w,v)\in A} f(w,v) - \sum_{w:(v,w)\in A} f(v,w) = \mu(v) \ \ \forall \ v \in V. \tag{3.11}$$

Here, the constraints 3.11 are known as the demand constraints for vertices $v \in V$ stating that the total flow leading into the vertex $v \in V$ minus the total flow leading out of it equals the demand of value $\mu(v)$ at that vertex. Here, if $\mu(v) = 0$ for $v \in V$ then the vertex v is neither a supply vertex nor a demand vertex but a transshipment vertex on $\mathcal{N}$.

A minimum cost flow problem defined on network $\mathcal{N}$ with only transshipment vertices; i.e., $\mu(v) = 0$ for all $v \in V$, and with arc demands $l(a) \geq 0$ for all $a = (v,w) \in A$, is called *minimum cost circulation (MCC) problem*. The problem asks to find the feasible circulation f that has minimum cost given by objective function in (3.10). In this problem the demand constraint 3.11 obviously reduces to the following form:

$$\sum_{w:(w,v)\in A} f(w,v) - \sum_{w:(v,w)\in A} f(v,w) = 0 \ \ \forall \ v \in V. \tag{3.12}$$

Here, since no exogenous flow is introduced into the network or no flow is extracted from it, all the flow circulates around the network. Notice that the input for minimum cost flow problem is the same as for minimum cost circulation problem, except that there are no demands $l(a)$, but instead, there are demands $\mu(v)$ for all $v \in V$. Thus, we can see that the minimum cost circulation problem and the minimum cost flow problem are equivalent to each other. However, we make use of minimum cost circulations with arc demands $l(a)$ zero for all $a \in A$, while using in solving dynamic version of evacuation planning problems.

Whether the minimum cost flow problem defined on network $\mathcal{N}$ exists and has a feasible solution can be determined by solving a maximum (static) flow problem on it. Introduce a source vertex s and a sink vertex d on $\mathcal{N}$. For each vertex v with $\mu(v) > 0$, add a "source" arc (s,v) with capacity $\mu(v)$, and for each vertex v with $\mu(v) < 0$, add a "sink" arc (v,d) with capacity $-\mu(v)$. Now, solve a maximum flow problem from s to d. If the maximum flow saturates all the "source" arcs, minimum cost flow problem is feasible, otherwise it is infeasible. For details see [2].

Theorem 3.4.2 *[26] A flow f on a network $\mathcal{N}$ is a minimum cost circulation if and only if there is no directed cycle $\mathcal{C}$ in the residual network $\mathcal{N}_f$ such that the sum of the cost around the arcs in $\mathcal{C}$ is negative.*

Theorem 3.4.2 has been used to transform a maximum static flow f on network $\mathcal{N}$ into a minimum cost flow in 'Cycle-Canceling' algorithm in [69]. The algorithm requires the following steps: Begin with any circulation f. A starting circulation can be computed using any maximum flow algorithm. Find a negative residual cycle $\mathcal{C}$ and cancel it by increasing the flow on each of its arcs by an amount equal to the capacity of $\mathcal{C}$ repeatedly until there are no negative residual cycles.

Goldberg and Tarjan [48] make a careful choice of next cycle to cancel in each iteration that leads to strongly polynomial algorithm known as 'Minimum-Mean Cycle-Canceling' algorithm. The selection rule is simple: Always cancel a residual cycle whose average arc cost is as small as possible. Mean cost of cycle is its cost divided by the number of arcs it contains. The minimum mean cycle-canceling algorithm runs in $O(n^2m^3logn)$ time. Goldberg and Tarjan [49] developed a new approach for solving minimum cost circulation problems by combining the method for solving the maximum flow problem with successive approximation techniques based on cost scaling. This 'cost-scaling' algorithm for minimum-cost circulation problem can be implemented in $O(n^3min(log(nC), mlogn))$ time, and is known to be one of the most efficient algorithms for the problem. However, its efficiency in practice depends on many implementation aspects [25].

For two terminal series parallel network, a Greedy Algorithm has been proposed for minimum cost flow problem with integer arc capacities and arc costs in [10]. We give a detailed description of this algorithm in Chapter 4.

3.5 Dynamic Flow Problems

An evacuee requires time for traveling from one place to another. Thus, it is necessary to take this travel time parameter into account while designing a mathematical optimization model of evacuation plan. Dynamic network is a tool for the model in which each arc has transit time, modeling the time taken by an evacuee to traverse the arc besides capacity that bounds the number of evacuees from above. Dynamic networks were first introduced while dealing maximum dynamic flow problem [42]. The model is based on discrete time setting.

Since then, several further problems have been analyzed, for example, the quickest flows, earliest arrival flows, flows with continuous time setting, or models where the parameters change with time, see [73].

In this section, we discuss dynamic network flow problems with three distinct objectives that stand as basis for modeling evacuations. The first is *maximum dynamic $s-d$ flow problem* that asks to send the maximum units of flow (evacuees) from the source (risk zone) to the sink (safety) within specified time horizon. The second is *quickest flow problem* that sends the given number of evacuees into the safety in minimum possible time. And, the third is *earliest arrival flow problem* that achieves the objective of the first problem at every time point within the time horizon. We also discuss solution techniques to these problems in brief. Extension of these problems will appear in chapters 4 and 5.

3.5.1 Maximum dynamic $s-d$ flow problem

Consider a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with source vertex s and sink vertex d. Then the objective of the *maximum dynamic $s-d$ flow problem* designed on $\mathcal{N}$ with continuous time setting is to maximize a net feasible continuous dynamic flow $\mathbf{f}_c$ within the specified time horizon T, from the source s to the sink d, given by

$$\mathbf{f}_c = \sum_{a\in\delta^-(d)} \int_{\xi=0}^{T-\tau(a)} f(a,\xi) - \sum_{a\in\delta^+(d)} \int_{\xi=0}^{T} f(a,\xi). \tag{3.13}$$

A maximum dynamic $s-d$ flow problem on network $\mathcal{N}$ is also known as *maximum dynamic flow problem* on single source and single sink network. In the case of discrete time setting, the objective of the problem on $\mathcal{N}$ is to maximize a net feasible discrete dynamic flow $\mathbf{f}_d$ within the specified time horizon T, from the source s to the sink d, given by

$$\mathbf{f}_d = \sum_{a\in\delta^-(d)} \sum_{\xi=0}^{T-\tau(a)} f(a,\xi) - \sum_{a\in\delta^+(d)} \sum_{\xi=0}^{T} f(a,\xi). \tag{3.14}$$

It is important to mention that the smaller the discrete time intervals are, the better the approximation. So, accuracy of discrete time models is a trade-off to computation time, as it takes more time to consider more time intervals. It is considered that, without loss of generality, $f(a,t) = 0$ for time t before and after time T and it is expected that all flow units leave the network within it.

From computational point of view, solutions to the flow problems modeled on dynamic network are more complex than static flow problems in the sense that the flow is specified for all time points $t \in [0, T)$. In the following, we discuss the solution techniques that solve maximum dynamic flow problem with constant transit times. In this thesis, we consider the evacuation problems modeled on network with constant transit times.

3.5.1.1 Time expanded network

Solution procedures developed for maximum static flow problem can also be applied to solve the problem modeled on dynamic network by using time-expanded network [42]. This special network acts as a static representation of flow variables defined on dynamic network where only integral time points are considered. For dynamic network $\mathcal{N} = (V, A, T)$, its time expanded network is an expansion of $\mathcal{N}$ over time horizon T. It is denoted by $\mathcal{N}^T = (V^T, A^T)$ and constructed as follows: Copy each vertex v of dynamic network $\mathcal{N}$ for T times, one for each time unit, so that vertices in $\mathcal{N}^T$ are of the form $v(t)\ \forall v \in V,\ t \in \{0, 1, \ldots, T\}$. Also, copy each arc (v, w) of network $\mathcal{N}$ between vertices v and w with transit time $\tau(v, w)$ and capacity $u(v, w)$, so that arcs in $\mathcal{N}^T$ are of the form $(v(t), w(t+\tau(v, w)))$ for all $v, w \in V,\ t \in \{0, 1, \ldots, T - \tau(v, w)\}$ with capacity $u(v, w)$. Here, copy of vertex v in time unit t and copy of vertex w in time unit $t+\tau(v, w)$ are denoted as $v(t)$ and $w(t+\tau(v, w))$, respectively. If storage of flow in vertices is allowed (essential sometimes for algorithmic view point), the holdover arcs are needed that connect copy of vertex v in time unit t to copy of vertex v at time unit $t+1$. Thus, the set of holdover arcs consists of all arcs $(v(t), v(t+1))\ \forall\ t \in \{0, 1, \ldots, T - 1\}$. Time-expanded network for discrete time setting has been depicted in Figure 3.4.

There is inter-relation between a static flow on time-expanded network and dynamic flow on the corresponding dynamic network. The relation can be viewed in the following sense: Assume we are given a static flow $f(a)$ on arcs a on $\mathcal{N}^T$. Each arc a on $\mathcal{N}^T$ corresponds to an arc, say a', on the original network $\mathcal{N}$. Let the tail of arc a be at time point t. Then dynamic flow on arc a' at time t has flow rate $f(a', t) := f(a)$. The feasibility of this flow follows by the construction of expanded network. From same interpretation, a discrete dynamic flow on $\mathcal{N}$ can be viewed as static flow on time-expanded network. Thus, a static flow on time-expanded network can directly be transformed into a discrete dynamic flow on the original network and vice versa. That is, if f_s is a maximum static $s(0) - d(T)$ flow on time-expanded network $\mathcal{N}^T$ then there is a maximum dynamic flow f_d on dynamic network $\mathcal{N}$ with time horizon T, such that $\mathbf{f}_s$ on $\mathcal{N}^T$ is equal to $\mathbf{f}_d$ on $\mathcal{N}$.

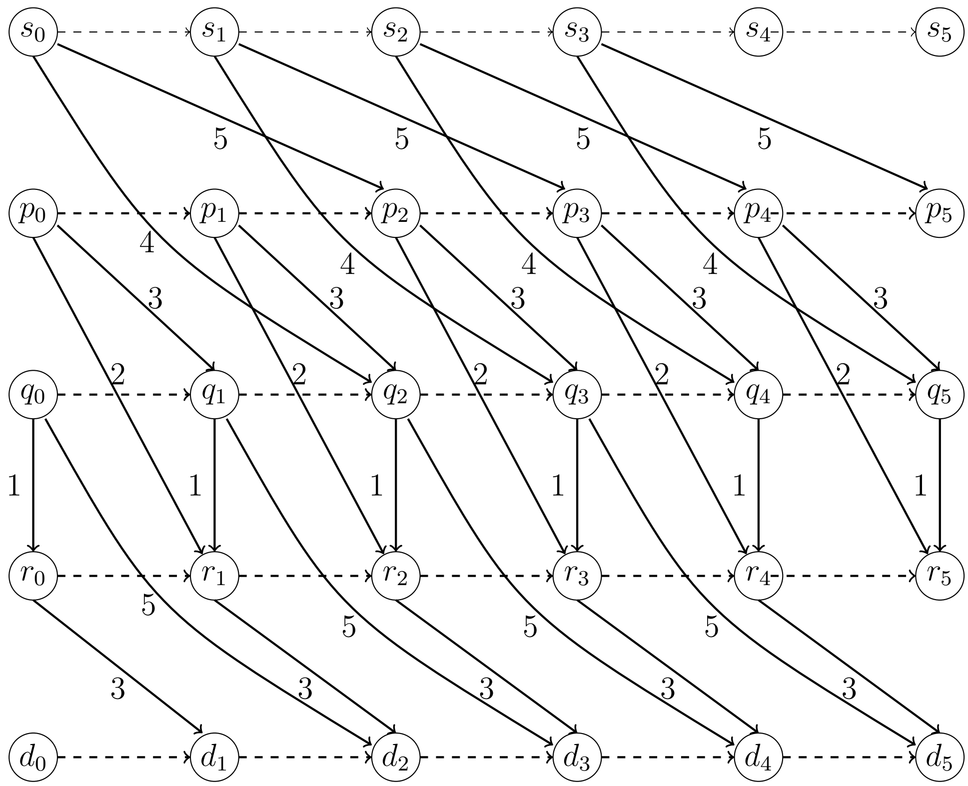


Figure 3.4: Time-expanded network with holdover arcs (dashed directed lines) for discrete time setting of the network depicted in Figure 3.1 for time horizon $T = 5$.

Notion of natural transformation, indicated by Fleisher and Tardos [40], is helpful to generalize a discrete dynamic flow on a network $\mathcal{N}$ as a continuous dynamic flow. The notion states that the amount of flow that arrives at vertex w through arc $a = (v, w) \in A$ at time step t in discrete time setting is equal to the amount of flow arriving at w through arc $a = (v, w)$ during unit interval of time $[t, t+1)$, i.e., $f_d(a, t) := f_c(a, [t, t+1)) \ \forall \ t \in \{0, 1, \ldots, T-1\}$. Here, capacity constraints for continuous dynamic flow f_c are obviously obeyed, since $f_d(a, t) \leq u(a)$ implies $f_c(a, [t, t+1)) \leq u(a)$ for all time points in the interval $[t, t+1)$. This transformation is a bidirectional, if T and all transit times are integral [7]. The notion can also be applied to interpret the flow computed on time-expanded network $\mathcal{N}^T$ as a continuous dynamic flow. Here, each time point $t \in \{0, 1, \ldots, T-1\}$ represents a unit time interval $[t, t+1)$. Then, a static flow on an arc with target vertex in time point t is now interpreted as flow arriving at this vertex during the time interval $[t, t+1)$. Equivalently, a static flow on an arc with starting vertex at time point t is now interpreted as the flow sent out of this vertex during the time interval $[t, t+1)$.

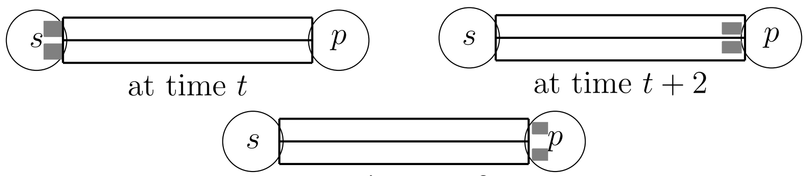


Figure 3.5: Continuous time flow on arc having capacity 2 and transit time 2.

Flow movement on arc (s, p) with continuous time setting has been shown in Figure 3.5 where the capacity $u(s, p) = 2$ and transit time $\tau(s, p) = 2$. Each unit of flow entering the arc at time t starts reaching at p at time $t + \tau(s, p)$, that is, at time $t+2$. Flow totally leaves the arc only at time $t + \tau(s, p) + 1$, that is, at time $t+3$. However, it can be considered that the same amount of flow completes its journey at time $t+2$ in discrete time setting.

Thus, time-expanded networks $\mathcal{N}^T$ seem to be a mighty tool to solve dynamic network flow problems. In addition to this, time-expanded networks are useful for proving the correctness of many network flow algorithms. The correctness of maximum dynamic $s-d$ flow algorithm can be proved, for example, by determining a cut in the corresponding time-expanded network [42]. Most of the algorithms presented in Chapter 4 have been shown correct with the help of these networks. However, limitation associated to this is that the size of transformed network depends linearly on time horizon T. Therefore, any algorithm that runs in polynomial time for a static network executes only in a pseudo-polynomial time on time expanded network $\mathcal{N}^T$ of the dynamic network $\mathcal{N}$. In the worst case scenario, T is exponential in the input size of the problem. Thus, the size of time expanded network makes the problem solution prohibitively expensive in many cases.

3.5.1.2 Polynomial solution

Maximum dynamic flow problem can also be solved efficiently using the minimum cost circulation (MCC) problem on given network. We start with the following definition of *temporally repeated flows (TRFs)* which is an intuitive concept for computation of maximum dynamic flows.

Definition 3.5.1 ***[Temporally Repeated Flows]*** *Let* $\mathcal{N} = (G, u(a), \tau(a), s, d, T)$ *be a dynamic flow network. Let* f_m *be the optimal (static) minimum cost flow on* $\mathcal{N}$*, and* $(\Gamma, f(\gamma))$ *be its corresponding flow decomposition. Then the temporally repeated dynamic* $s-d$ *flow with time horizon* T *induced by* f_m*, denoted by* Γ^T*, is defined as follows: Send the flow on each path* $\gamma \in \Gamma$ *in a constant rate of* $f(\gamma)$ *at time zero and continue on them as long as there is enough time left within time horizon* T *to arrive at the sink* d*. Thus, if* $\tau(\gamma)$ *denotes the sum of the transit times of arcs on path* γ*, the value of* Γ^T *for discrete time setting is denoted by* $\boldsymbol{f}_d$ *and is given by*

$$\boldsymbol{f}_d := \sum_{\gamma \in \Gamma} f(\gamma)(T + 1 - \tau(\gamma)).$$

And, the value of Γ^T *for continuous time setting is denoted by* $\boldsymbol{f}_c$ *and is given by*

$$\boldsymbol{f}_c := \sum_{\gamma \in \Gamma} f(\gamma)(T - \tau(\gamma)).$$

An optimal solution to minimum cost circulation (MCC) problem can be turned into a maximal discrete dynamic flow using the notion of temporally repeated flows [42]. The idea of the procedure (see Algorithm 3.5.1) is quite simple. First compute a minimum cost circulation on the network $\mathcal{N}$, decompose it into $s-d$ paths, and compute temporally repeated flow. To set the MCC problem on $\mathcal{N}$, interpret the transit times $\tau(a)$ as cost coefficients $c(a)$ for each arc $a \in A$. Also, it is desired to maximize dynamic flow on $\mathcal{N}$ so that the cost of circulation is minimum and that all flow arrives at the sink within time T. Thus, it is required to model time horizon T in the solution techniques of MCC problem to be allowed to transfer the results of this problem to maximum dynamic flow problem. For this, an additional arc (d, s) from the sink d to the source s with sufficient capacity and transit time $(T + 1)$, is inserted on original network $\mathcal{N}$. Then compute a minimum cost circulation f_m on the extended network. Note that the flow along inserted arc (d, s) is ignored. Thus, we can compute a temporally repeated flow Γ^T with maximum value, $\mathbf{f}_d$ or $\mathbf{f}_c$ depending upon the flow model, by a minimum cost flow computation on network $\mathcal{N}$.

Due to feasibility of minimum (static) cost circulation f_m and nature of its path decomposition, the flow generated by temporally repeated flow is a feasible dynamic flow on $\mathcal{N}$. Any temporally repeated solution corresponding to a feasible static circulation f_m is a feasible dynamic flow, and there exists an optimal solution to maximum dynamic flow problem that is in a temporally repeated structure [42]. Also, by the construction of temporally repeated flow, there is no flow on network before time zero and after time T.

Algorithm 3.5.1 MCC Algorithm [42]

1. Given a network $\mathcal{N} = (G, u(a), \tau(a), s, d, T)$ with additional arc (d, s) where $u(d, s) = \infty$ and $\tau(d, s) = -(T + 1)$.

2. Compute a minimum cost circulation f_m on $\mathcal{N}$ with transit time of each arc as cost coefficient.

3. Compute a flow decomposition $(\Gamma, f(\gamma))$ of f_m.

4. Find a temporally repeated flow Γ^T on $\mathcal{N}$.

It is important to mention that there does not exist a deterministic polynomial time algorithm that solves dynamic minimum cost $s - d$ flow problem with given supply, see [70]. For this problem, a dynamic flow is to be computed that sends the total supply into the sink within a given time horizon T. However, the corresponding static flow problem is easy to solve as we saw above.

3.5.2 Quickest flow problem

Depending upon evacuation situation one may be interested in asking for a minimum possible time horizon for which a given number of evacuees at risk zone could be sent to safe zone instead of asking for a maximum number of evacuees within a given time horizon. The scenario can be modeled as a *quickest flow (QF) problem.*

Consider a single-source-single-sink dynamic network $\mathcal{N} = (G, u(a), \tau(a), s, d)$ with given demand μ. Then a quickest flow problem on $\mathcal{N}$ asks for a minimum possible time horizon T_μ for which a dynamic $s - d$ flow of given value μ exists. That is, a quickest flow problem on dynamic network $\mathcal{N}$ accomplishes of sending the pre-assigned flow units of value μ from the source s to the sink d in minimum possible time horizon T_μ by respecting capacity constraints (3.3) and flow conservation constraints (3.4). Burkard et al. [24] showed that QF problem is closely related to MDF problem by computational point of view. However, the problems seem to be inverse of each other with respect to their objectives.

In the following we make a brief sketch of a naive solution technique proposed in [24] for QF problem. In this method, for an strictly increasing sequence of integral time points $\{T_n\}$, an initial time interval $I_0 = [T_l, T_u]$ such that $\mathbf{v}(T_l) < \mu < \mathbf{v}(T_u)$, is taken. Here, $\mathbf{v}(T_l)$ and

$\mathbf{v}(T_u)$ denote maximum dynamic flow value for time horizon T_l and T_u, respectively. Clearly, $T_l \leq T_\mu \leq T_u$ where T_μ is the minimum time that requires for flow units of value μ to send from the source s to the sink d. Then the mid-point, say, T_m, of the interval I_0 is computed and $\mathbf{v}(T_m)$ is checked for whether it is equal to, less than or greater than μ. Depending upon this value, it is decided whether the procedure ends, or should work on the next interval on left or right of the mid-point T_m.

Moreover, due to the nature of construction of maximum flow using this technique, the maximum flow of value μ obtained for time horizon T, could also be possible to find in lesser time horizon T'. That is, it cannot be guaranteed that the time horizon T within which the dynamic flow of value μ can be sent to the sink d is the minimum to attain this flow value. One should check whether the same flow value is attained for some lesser time point T'.

Thus, major step in the procedure above is to perform a binary search over time horizon T and to solve a maximum dynamic flow problem in each iteration until the minimum time needed to send given amount of flow μ is found. Authors in [24] applied Newton's method to improve binary search method and obtained a time bound of $O(log(nU)\, MCF(n,m))$ where n, m and U are number of vertices, arcs and the maximum arc capacity, respectively, and $MCF(n,m)$ is the time bound for solving minimum cost flow problem. The time T of the quickest flow is a rational number with a denominator bounded by the size of a minimum cut in network, if μ is an integer and transit times are integral, and thus T can also be computed by binary search for the QF problem with continuous time setting [40].

Authors [24] are also able to give a strongly-polynomial time algorithm for quickest flow problem using Megiddo's parametric search [79] which runs in time $O(m^2\, log^3 n(m+n\, logn))$. Here, it needs to repeatedly call minimum cost flow problem as a subroutine. Therefore, despite of being quite similar with maximum dynamic flow problem in nature, it is not possible to solve the quickest flow problem within the same time bound as the minimum cost flow problem using the technique discussed here.

The question whether it is possible to develop an algorithm that solves quickest flow problem within the same time bound as minimum cost flow problem has been affirmatively answered by Lin and Jaillet [75]. They reformulated the quickest flow problem as a parametric minimum cost flow problem. They extended cost-scaling algorithm introduced in [49] to design a new cost-scaling algorithm that solves quickest flow problem with integer arc costs. Since cost-scaling algorithm remains one of the fastest for solving minimum cost flow problem in terms of the worst-case time bound (see [25]), their algorithm to solve quickest flow problem runs within the same time bound as one of the best algorithms for minimum cost flow problem. Their result also shows that unless quickest flow problem can be shown to be simpler than the minimum cost flow problem, their algorithm will remain one of the fastest for solving quickest flow problem. For C being maximum cost (travel time, in fact) on arcs, the extended cost-scaling algorithm runs in $O(nm\, log(n^2/m)\, log(nC))$ time by using dynamic tree data structure [95].

Quickest Path Problem. In some evacuation problems, given number of evacuees should be sent along only one path. This leads an evacuation problem to a *quickest path problem*, a variation of QF problem, in which we ask for an $s - d$ path through which we can send a given amount of flow as quickly as possible. This problem can be solved in polynomial time, see [27], [90] and [76].

3.5.3 Earliest arrival flow problem

Another network flow problem related to maximum dynamic flow problem is *earliest arrival flow (EAF) problem*. For the same network architecture and pre-specified time horizon T as described in the case of maximum dynamic flow problem, objective of EAF problem is to maximize flow units entering into the sink d at each time step $t \in [0, T)$. In addition, if flow should leave the source as late as possible, then the problem is known as *universally maximum dynamic flow problem.* Earliest arrival flow problems are of much interest to evacuators because these problems ensure that the number of evacuated persons is maximum at each time point within given time horizon T. This is desired during emergency evacuations since it is usually not known in advance how long the affected buildings or other structures can resist themselves before complete collapses.

Existence of solution to earliest arrival flow problem was proved by Gale [44]. His main contribution is existence proof of earliest arrival flows with three different considerations. The first is single-source-single-sink dynamic network. The second is single-source-single-sink dynamic network with time dependent capacities and transit times. And, the third is multiple-sources-single-sink dynamic network with time dependent capacities and transit times. He pointed out that the existence theorem does not extend to the network with multiple sinks.

To the best of our knowledge no polynomial time exact algorithm for earliest arrival flow problem on general network is known till now. Minieka [80] proposed an approach that allows some interesting insight into the structure of arrival and departure patterns of maximal flows. The patterns have been developed by introducing the concept of lexicographically maximum flows on static networks with multiple sources and sinks. These patterns can be translated to dynamic flows which have earliest arrival property by using time expanded networks. This procedure yields an exact solution to earliest arrival flow problem which has pseudo-polynomial time complexity since it depends directly on T. A similar exact algorithm with pseudo-polynomial time complexity is due to an independent work of Wilkinson [103]. There exists a fully polynomial time approximation algorithm devised by using the notion of *generalized temporally repeated flows* induced from *non-standard path decomposition* of flow to find earliest arrival flows which gives $(1 + \epsilon)$ approximation for any fixed $\epsilon > 0$ [57]. The path flows may use oppositely directed flows on arcs in the case of non-standard path decomposition. There exists an algorithm, working for discrete as well as continuous time model, that is polynomial in the input plus output size for multiple-sources-single-sink network, [9]. An algorithm for continuous time model is proposed in [40]. A fully polynomial-time approximation algorithm for continuous time model is proposed in [39]. Earliest arrival flow problem in network with multiple sinks has been studied in [92] where all arc transit time are zero. For this setting, they have given a complete characterization of the class of networks that always allow for earliest arrival flows. An earliest arrival flow problem, maximizing the ratios of flow values to capacities on the sinks lexicographically instead of strictly obeying the capacity constraints on them, has been studied in [61]. A pseudo-polynomial and a polynomial time algorithms for solving the problem with arbitrary and zero transit time for every arc, respectively have also been proposed.

Due to definition of earliest arrival flows, there is an obvious observation that all earliest arrival flows are maximum dynamic flows but converse is not necessarily true, see Example 4.5.2. Steiner [96] proved, by counter example, that for a general network $\mathcal{N} = (V, A, T)$ it is not always possible to find an earliest arrival flow which is temporally repeated flow. Ruzika et al. [91] (cf. [96]) considered maximum dynamic flow problem on special class of networks known as two terminal series parallel (TTSP) network. They developed a polynomial time MDF algorithm for TTSP networks and showed that this also solves earliest arrival flow problem for these networks. Authors also pointed out that on series-parallel networks it is always possible to find an earliest arrival flow as a temporally repeated flow. However, the converse is not necessarily true.

In this thesis, we consider earliest arrival flow problems in two terminal series parallel networks. The procedure for solving the problem on these networks has the following major steps: Develop an algorithm that computes a maximum dynamic flow on special structure of series-parallel networks. For this, it is necessary to analyze the minimum cost flow problem for series-parallel networks. We explain detailed solution idea of earliest arrival flow problem defined on TTSP network in Chapter 4. We study lexicographic version of the problem for uniform path length two terminal series parallel network in Chapter 5.

Multi-commodity flow problem. The flow model discussed above is single-commodity flow model where transshipment vertices are in uniform status, meaning that these vertices transfer the single commodity through them. Multi-commodity flow problems arise when more than one flow entities travel through the same network. Such flows may have different sources and sinks, i.e., certain sources can ship only to certain sinks, and satisfy the flow conservation constraints separately. However, the flows are to be bounded together by the common capacity constraints. Multi-commodity flow problems are much harder to solve than single-commodity problems [73]. These problems are beyond the scope of this thesis.

CHAPTER 4

EVACUATION PLANNING PROBLEMS WITH CONTRAFLOW APPROACH

4.1 Introduction

The reversibility of direction of traffic flow in one or more lanes of roadways for fixed time period is termed as contraflow. The contraflow approach reconfigures the network identifying ideal direction and reallocating available capacity for each arc to improve the flow egress time and/or the number of flow units from source to destination. The approach, due to its lane-direction reversal property, can be taken as a potential remedy to mitigate congestion during emergencies by increasing outbound evacuation route capacity. It significantly reduces the total evacuation time and/or increases the number of evacuees sent from risk zone to safety. Studies show that reversing one lane of a four-lane dual highway increases the evacuation road capacity by approximately 30% and reversing all the inbound lanes, it increases by 67% [102].

Despite the long history of studies on contraflow approach, there is limited implementation in real emergency evacuations due to difficulty in using commonly employed methods to duplicate traffic conditions of real contraflow lane during an emergency [102]. However, they have been adapted, in recent years, for evacuating major metropolitan regions threatened by disasters. It was first applied to the evacuation during Hurricane Floyd in the United States in 1999 with mixed, though overall positive, results [101]. Contraflow was also implemented during hurricanes Katrina and Rita in the United States in 2005. However, it was criticized as unplanned contraflow orders and as failure to use contraflow lanes [68]. Contraflow approach is primarily important for emergency evacuations, nonetheless, its applications are not limited to these. This is commonly used for accommodating directionally imbalanced traffic associated with daily commuter in big cities as well as consequences due to religious gathering, arrangement of concerts or tournaments, etc.

Numerous contraflow models: mathematical optimization, simulations or mixed models, for the evacuation tasks with different road network behaviors have been proposed in the literature. However, an obvious enumeration procedure for them is quite challenging due to its very large search space and high computational costs for even a small size network. We focus on the mathematical optimization contraflow models in this thesis. Since time parameter plays a vital role in designing evacuation planning models, it is important to be careful about its nature: discrete or continuous, adapted in the model. Optimization contraflow models developed so far are based on equal transit time settings on anti-parallel arcs and these models do not allow multiple arcs of different transit time. We call the two directed arcs *'anti-parallel'* if they are between the same pair of vertices, but in opposite directions. It is crucial, in case of uneven road architecture for example, to take contraflow models over multi-network into account for preparing evacuation tasks. Multi-networks capture the situation of roads with parallel lanes of different transit time, and obviously, anti-parallel arcs with not necessarily equal transit time. Moreover, it is crucial to consider whether transit time parameter behaves symmetrically during the reversal of lane direction. However, in the models presented in this thesis, it has been assumed that the transit time parameter behaves symmetrically during the reversal of direction of arcs. We consider discrete as well as continuous aspect of transit time parameter in the model, whereas capacities and transit time on arc are considered to be time independent.

We sketch a brief history on models and results of network contraflow evacuation planning problems in Section 4.2. Section 4.3 presents static contraflow problems. Dynamic contraflow problems with equal transit time on anti-parallel arcs have been presented in Section 4.4 and that the problems modeled on multi-network in Section 4.5.

4.2 Literature on Network Contraflow Problems

There is about two decade long history on study of network contraflow problems. Two evacuation algorithms: '*all-links*' and '*fastest-links*', have been proposed to deal contraflow evacuation planning problems, and performed simulation computations to compare their performances in [54]. This model does not take overall capacity of road network into account, but permits lane reversals with its partial capacities. An optimization (mesoscopic indeed) model has been considered for contraflow problem without scalable experiments based on dynamic traffic assignment method in [98]. Authors [67] studied contraflow evacuation planning problem, aiming to minimize evacuation egress time, for the first time using network flow theory. They highlighted the need of looking for a good solution rather than an exact optimal one for practical cases and proposed two heuristic solutions: '*flip high flow edge*' and '*simulated annealing (SA)*', to their problem. In the first, they solve a minimum cost problem on time-expanded static network for given time horizon to record the flow history and flip direction of each arc in favor of the direction of larger flow. The second heuristic yields a local minimum with evacuation egress time as the objective function, and random flipping based perturbations. A Tabu-based heuristic approach, which is a search-based iterative optimization technique for the problem, has been proposed in [99]. Due to the combinatorially increasing number of candidate networks reconfigured by contraflow, the Tabu search method may lead to a large search space in cases of very large spatial networks. The problem has been modeled as an integer programming formulation and two heuristics have been investigated as solution techniques to the problem in [68]. One of them is Greedy heuristic which determines the condition of congestion and flips highly congested arc in a greedy manner, and the other is Bottleneck Relief heuristic which identifies the bottleneck of network and increases its capacity by contraflow to improve maximum flow in each iteration. The Greedy heuristic produces high-quality solutions with significant performance, and the Bottleneck Relief heuristic is suitable to deal with large size of problem. However, their solution is based on empirical results. For more insight on contraflow problems, we refer to survey article [30].

The first mathematical optimization model for contraflow problem is due to Rebennack et al. [89]. They have investigated the analytical solution of maximum contraflow problem with polynomial time complexity for both static and dynamic networks. Their solution idea is based on transformation of input network into one at which the existing algorithms are applicable. They also showed that dynamic transshipment contraflow problem is NP-complete in the strong sense and contraflow problem with fixed switching cost is NP-hard even for static case. Thus, general contraflow problems are computationally hard. However, maximum contraflow problems can be solved in strongly polynomial time. For detailed solution procedures of these problems, see Subsections 4.4 and 4.5.

Quickest contraflow (QCF) problem on single-source-single-sink network has also been solved polynomially in [89]. The solution is based on parametric search algorithms of [24]. Earliest arrival contraflow (EACF) problem for TTSP network has been studied in [31] and proposed a polynomial time solution. For solution procedure, see Subsection 4.4.1. For multiple-sources-multiple sinks general networks with given supply and demand, lexicographically maximum dynamic contraflow problem has been solved polynomially in [86]. Contraflow approach has been incorporated in network flow model to study facility location problem in [32], and notion of abstract flow has been applied to network contraflow problems in [33]. The partial contraflow approach on abstract network setting has been introduced in [88]. Evacuation planning problem with time minimization objective has been studied for integrated network, composed of pickup network with partial lane reversibility and transit vehicle assignment network, in [1].

The dynamic contraflow problems studied so far are with discrete time setting and with consideration of equal transit time on anti-parallel arcs. As a contribution of this thesis, we study MDCF problem for continuous time setting and propose an efficient solution procedure for it (see Subsection 4.4.2). We also study MDCF problem and EACF problem modeled on ordinary network with not necessarily equal transit time on anti-parallel arcs as well as on multi-network, and propose efficient solution procedures to them with discrete as well as continuous time setting, see Section 4.5.

4.3 Static Contraflow Problems

In this section we introduce contraflow problems over static network and discuss its solution approach. Result and solution idea of these problems turn out to be useful in optimality proofs of solutions to problems modeled on dynamic network in successive sections.

Given a static network $\mathcal{N} = (V, A, u(a), s, d)$ with the source $s \in V$, the sink $d \in V$ and capacity $u(a)$ on each arc $a \in A$. Then an $s-d$ contraflow problem modeled on $\mathcal{N}$ maximizes a feasible static $s-d$ flow on $\mathcal{N}$, if direction of arcs can be reversed. The problem is also known as a *maximum static contraflow (MSCF) problem* on network $\mathcal{N}$ with single source and single sink.

Now we discuss the solution procedure to MSCF problem proposed in [89]. The procedure is based on modification of input network into a new network by summing the capacities on arcs (v, w) and (w, v) such that MSCF problem reduces to MSF problem on it. In particular, the procedure has following steps: At the first, given static network $N = (V, A, u(a), s, d)$

is transformed into its auxiliary network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), s, d)$ where the arc set $\tilde{A}$ contains undirected arc (v, w), if (v, w) and/or (w, v) belong to original arc set A with capacity $u(\widetilde{v,w}) = u(v, w) + u(w, v)$. At the second, a maximum static $s - d$ flow is computed on so-formed undirected network $\tilde{\mathcal{N}}$ by using any known algorithm. It is not known in advance that which of the arcs (v, w) or (w, v) is to be flipped until maximum flow is computed on modified network. There exists an optimal flow to maximum flow problem that does not have cycles. Here, to ensure this property, the computed maximum static flow is decomposed into set of paths from source to sink and set of cycles with positive flows and cancel positive flows along all cycles. Thus, arcs on both direction will never be used in this flow for maximum flow problem on $\tilde{\mathcal{N}}$. This ensures that the constructed solution to maximum static contraflow problem is well defined. The solution procedure has been summarized in Algorithm 4.3.1.

Algorithm 4.3.1 MSCF Algorithm [89]

---

1. Given a static network $\mathcal{N} = (V, A, u(a), s, d)$ with integer inputs.
2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), s, d)$ where

$$\tilde{A} = \{\tilde{a} = (v, w) : (v, w) \in A \vee (w, v) \in A\},$$
$$\text{and, for all } \tilde{a} \in \tilde{A}, \text{ the capacity } u(\tilde{a}) := u(v, w) + u(w, v).$$

3. Solve the maximum flow problem on network $\tilde{\mathcal{N}}$.
4. Perform the flow decomposition into path and cycle flows of the maximum flow obtained from step-3 and remove all cycle flows.
5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
6. Obtain MSCF solution on $\mathcal{N}$.

We state following lemma that shows equivalence between the optimal flow on $\tilde{\mathcal{N}}$ and the optimal contraflow on input network $\mathcal{N}$ and turns out to be useful in optimality proof of algorithms designed for contraflow problems on dynamic networks in this thesis.

Lemma 4.3.1 [89] *The maximum static contraflow problem on a static network $\mathcal{N}$ is equivalent to the maximum static flow problem on the corresponding transformed network $\tilde{\mathcal{N}}$.*

Time complexity of the solution procedure (Algorithm 4.3.1) depends on time complexity of the solution procedure on the reduced network $\tilde{\mathcal{N}}$. Also, it is dominated by solving a maximum flow problem on $\tilde{\mathcal{N}}$ and by the flow decomposition, since network transformation can be done only in $O(m)$ time. If one wishes to apply highest-label preflow-push algorithm having complexity $O(n^2\sqrt{m})$ of [28] and flow decomposition algorithm having complexity $O(mn)$ of [2], the procedure runs in strongly polynomial time.

Maximum contraflow problem for static network $\mathcal{N}$ with multiple sources and multiple sinks with respective surplus and deficits, known as *maximum static transshipment contraflow (MSTCF) problem*, is also polynomially solvable [89]. This can be realized from the network modification idea in which multiple sources and multiple sinks are treated as single source and single sink such that the MSTCF problem turns into a static $s-d$ contraflow problem. For the network modification procedure, we refer to Subsection 3.4.1.

4.4 Dynamic Contraflow Problems

4.4.1 Discrete time dynamic contraflow problems

The study of dynamic network flow problems is an interesting as well as challenging field of optimization. In contrast to static flow problems, they include a temporal dimension and consequently provide a more realistic modeling tool for a wide variety of applications. Of course, the time parameter plays a crucial role in evacuation planning problem. Thus, dynamic version of contraflow problems are crucial. The first optimization model for the discrete dynamic contraflow problems has been given in [89]. In the following we discuss maximum dynamic contraflow problem, quickest contraflow problem and earliest arrival contraflow problem with their solution techniques with the details.

4.4.1.1 Discrete time maximum dynamic contraflow problem

Given a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with single source $s \in V$, single sink $d \in V$, capacity $u(a)$ and transit time $\tau(a)$ on each arc $a \in A$ with $\tau(v,w) = \tau(w,v)$ if $(v,w), (w,v) \in A$ and pre-specified time horizon T. Then the objective of a maximum dynamic $s-d$ contraflow problem on $\mathcal{N}$ is to maximize net feasible discrete dynamic flow $\mathbf{f}_d$ from s to d within T, if direction of arcs on $\mathcal{N}$ are allowed to reverse. For T being discretized, this problem is also known as *discrete time maximum dynamic contraflow (DT-MDCF) problem* on $\mathcal{N}$ with single source and single sink. This definition states that in a MDCF problem the capacities on anti-parallel arcs could be unequal on $\mathcal{N}$, whereas the transit time must be the same. This consideration on the model says that the flipping of an arc only changes the capacities of the arcs on modified network but does not alter their transit time. Moreover, it is to be noted that, if we choose to flip an arc, it remains flipped from time 0 to T.

Now we discuss the solution procedure for DT-MDCF problem proposed in [89]. As in the case of MSCF problem in Section 4.3, the procedure is based on reduction of input network into new network on which ordinary MDF problems can be solved efficiently. The procedure has following two major steps: At the first, given dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ is transformed into its auxiliary network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ where arc set $\tilde{A}$ contains undirected arc (v, w), if $(v, w) \in A$ and/or $(w, v) \in A$ such that capacity $u(\widetilde{v,w}) := u(v, w) + u(w, v)$ and transit time $\tau(\widetilde{v,w}) := \tau(v, w)$ if $(v, w) \in A$, otherwise, $\tau(\widetilde{v,w}) := \tau(w, v)$. At the second, a maximum dynamic $s - d$ flow is computed on so-formed network $\tilde{\mathcal{N}}$. The maximum dynamic $s - d$ flow on $\tilde{\mathcal{N}}$ is obtained by using temporally repeated flows (TRFs) on it. The TRF is a dynamic flow that can be generated by repeating all possible source to sink path flows starting at time zero and then adopting temporal repetition as far as possible. For details about TRFs, see Subsection 3.5.1. The following theorem shows the importance of TRFs in the context of single-source-single-sink network flow problem as we consider throughout the thesis.

Theorem 4.4.1 [43] *There is a temporally repeated dynamic flow that is maximal over all dynamic flows for T periods.*

During the procedure we also decompose flow resulting from the application of minimum cost flow algorithm on $\tilde{\mathcal{N}}$ into path and cycle flows, and omit all cycle flows. Thus, one more reason to realize the importance of temporally repeated flows on the procedure is because it ensures that only one of the arcs (v, w) or (w, v) is used in constructed flows. The procedure for solving a DT-MDCF problem has been summarized in Algorithm 4.4.1.

A feasible dynamic flow on $\mathcal{N}$ has an equivalent feasible static flow on the corresponding time expanded network $\mathcal{N}^T$ [42]. This assures that a dynamic flow problem can be solved by converting dynamic network $\mathcal{N}$ into time expanded static network $\mathcal{N}^T$ over time horizon

Algorithm 4.4.1 DT-MDCF Algorithm [89]

1. Given a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ where $\tilde{A} = \{\tilde{a} = (v, w) : (v, w) \in A \vee (w, v) \in A\}$, and for all $\tilde{a} \in \tilde{A}$, capacity $u(\tilde{a}) := u(v, w) + u(w, v)$ and transit time $\tau(\tilde{a}) := \tau(v, w)$ if $(v, w) \in A$ and $\tau(\tilde{a}) := \tau(w, v)$ otherwise.
3. Compute dynamic, temporally repeated flow on network $\tilde{\mathcal{N}}$.
4. Perform flow decomposition into path and cycle flows of the maximum flow obtained from step-3 and remove all cycle flows.
5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
6. Obtain DT-MDCF solution on $\mathcal{N}$.

T. Similarly, every feasible flow to maximum dynamic contraflow problem on network $\mathcal{N}$ has an equivalent feasible flow to the maximum static contraflow problem of corresponding time expanded network $\mathcal{N}^T$. That is, the following lemma holds true.

Lemma 4.4.1 [89] *The maximum flow for two terminal case of the maximum dynamic contraflow problem on $\mathcal{N}$ does not exceed the optimal flow for the corresponding time expanded network $\mathcal{N}^T$.*

Feasibility of the solution computed by Algorithm 4.4.1 can be assured by checking whether arc reversal process is well defined. Here, the constructed flows are temporally repeated, and therefore, flow travels along only one direction between two vertices and never in both directions at the same time or different time steps. This is possible due to cancellation of positive flows along all cycles, which have been obtained by the decomposition of minimum cost flow on $\tilde{\mathcal{N}}$ into $s - d$ flows and cycle flows. Thus, every feasible flow of MDF problem on $\tilde{\mathcal{N}}$ is feasible to MDCF problem on $\mathcal{N}$. On the other hand, due to Lemma 4.3.1 and Lemma 4.4.1, MDCF on $\mathcal{N}$ is equivalent to MSF on $\tilde{\mathcal{N}}^T$. Also, by Theorem 4.4.1, this MSF can be obtained by temporally repeating an $s - d$ flow on $\tilde{\mathcal{N}}$. Thus, from these arguments, the following theorem can be established.

Theorem 4.4.2 [89] *Algorithm 4.4.1 solves the maximum dynamic contraflow problem for network $\mathcal{N}$ optimally.*

Overall time complexity of Algorithm 4.4.1 is dominated by the complexity of finding temporally repeated flow on $\tilde{\mathcal{N}}$ which is equivalent to solving a minimum cost flow (MCF) problem. The minimum mean cycle-canceling algorithm [48], for instance, leads to a strongly polynomial time of order $O(n^2\, m^3\, logn)$ for solving the MCF problem. Flow decomposition and network transformation requires same time complexity as in the case of static contraflow problem. Thus, Algorithm 4.4.1 runs in strongly polynomial time.

4.4.1.2 Discrete time quickest contraflow problem

A dynamic network flow problem closely related to maximum dynamic flow problem is the quickest flow problem that accomplishes of sending the given excess of flow units from the source to the sink in minimum possible time horizon T. See Subsection 3.5.2 for details. A *quickest contraflow (QCF) problem* finds a feasible dynamic flow f, if exists, on a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d)$ from the source s to the sink d which sends the given excess, say, μ from the source s to the sink d in the minimum time horizon T, if direction of arcs on $\mathcal{N}$ can be reversed. The QCF model is suitable during emergency if the number of evacuees at risk zone is known in advance and one wishes to shift all of them to the *safety* as quickly as possible, if direction of lanes are reversible.

Discrete time quickest contraflow (DT-QCF) problem can be solved optimally in a strongly polynomial time [89]. The solution idea is similar to the idea of [24] for solving a quickest flow problem. For solution, at the first, obtain an upper bound on the quickest time which can be obtained in polynomial time by applying MCC Algorithm 3.5.1, for instance, to compute $s - d$ paths, and temporally repeating flow along paths until all supply at the source s is sent to the sink d. Then, perform a parametric search method suggested in [79] by repeatedly solving the maximum dynamic contraflow problem on $\mathcal{N}$ using Algorithm 4.4.1. Complexity of the DT-QCF problem can be improved further by applying new cost-scaling algorithm for the quickest flow problem [75]. This algorithm runs with time complexity of order $O(nm\, log(n^2/m)\, log(nC))$ for C the maximum arc cost, if dynamic tree data structure [95] is applied.

4.4.1.3 Discrete time earliest arrival contraflow problem

Dynamic network flow problem that aims to maximize flow from source to sink in the basis of as to the earliest within pre-specified time bound, known as *earliest arrival flow (EAF) problem*, is of much interest to evacuators. There could be limited time bound, during disasters, to take evacuees out from risk zone to safer. Moreover, since it is usually not known in advance when the disaster will actually happen, it is desirable to design mathematical model for evacuation in such a way that as many evacuees as possible are saved. Given a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with single source $s \in V$, single sink $d \in V$, capacity $u(a)$ and transit time $\tau(a)$ on each arc $a \in A$ with $\tau(v, w) = \tau(w, v)$ if $(v, w), (w, v) \in A$ and pre-specified time horizon T. Then an EACF problem answers the question: What is the maximum flow from the source s to the sink d on $\mathcal{N}$ in each time step $t \in \{0, 1, \ldots, T\}$, if direction of arcs on $\mathcal{N}$ can be reversed?

To the best of our knowledge, there is no polynomial time procedure that gives exact solution for EAF problem with and without arc reversal capability designed on general network. However, there exists exact solution to the problem of both types for two terminal series parallel (TTSP) network, see [96], [91] and [31]. The DT-MDCF algorithm (Algorithm 4.4.1) can be modified making use of minimum cost circulation Algorithm 4.4.3 on TTSP networks. As a result, a maximum dynamic contraflow solution (Algorithm 4.4.4) on TTSP networks having earliest arrival property is obtained. Here, we discuss the solution procedure of the problem without arc reversal before presenting solution technique to the problem with arc reversal on TTSP network.

Let us recall two terminal series parallel (TTSP) network defined in Section 3.2. The following property plays a crucial role in computing minimum cost solution on TTSP network having earliest arrival property by avoiding negative cycles [91]. Let $P[v, w]$ be sub-network of series-parallel network $\mathcal{N}$ including all paths from the vertex v to the vertex w on $\mathcal{N}$. In this representation $P[v, w]$ corresponds to the smallest subtree including all arcs a with $head(a) = w$ and $tail(a) = v$. Thus, $P[v, w]$ is the inclusion-wise maximal (series or parallel) composition having v and w as its terminals. Any $s - d$ path on $\mathcal{N}$ using an arc of this sub-network $P[v, w]$ must use both vertices v and w.

Bein et al. [10] presented a greedy algorithm (Algorithm 4.4.2) that computes minimum cost flow on TTSP networks for all flow values up to the maximal flow value, say f_m. The algorithm starts with zero flow, iteratively finds the currently available cheapest path γ_k

Algorithm 4.4.2 MCF Algorithm for TTSP Network [10]

Input: TTSP network $\mathcal{N} = (V, A, u(a), c(a), s, d)$ with integer inputs.
Output: Minimum cost flow solution f_m with paths γ_k with corresponding cost $c(\gamma_k)$ and flow value $f(\gamma_k)$.
for all $(v, w) \in A$ **do**
$f(v, w) := 0,\ k := 0$
end for
while there exists a path connecting s and d on $\mathcal{N}$ **do**
$k := k + 1$
Find a minimum cost path γ_k and the corresponding cost $c(\gamma_k)$.
$f(\gamma_k) := min\,\{u(v, w) : (v, w) \in \gamma_k\}$
for all $(v, w) \in \gamma_k$ **do**
$u(v, w) := u(v, w) - f(\gamma_k)$
if $u(v, w) = 0$
then $A := A - \{(v, w)\}$
end if
end for
end while

with cost $c(\gamma_k) = \sum_{a \in \gamma_k} c(a)$ where $c(a)$ denotes the cost on arc $a \in A$ and assigns flow value $f(\gamma_k)$ to it in a greedy manner. Such path is calculated as the maximal residual capacity on the arcs of the path. Then, capacities on the arcs of γ_k are updated as $u(a) := u(a) - f(\gamma_k)$ for all $a \in \gamma_k$. This is an augmenting path based algorithm which does not use backward arcs. That is, once the algorithm sends flow along a path γ_k, this flow will never be taken back again. The cheapest path capacity has been totally used by this successive shortest paths computing algorithm before sending the flow along any other expensive path. The algorithm can be implemented in strongly polynomial time of order $O(nm + m\,logm)$ for all flow values up to the maximum flow value f_m where the minimum cost paths γ_k and its corresponding cost value $c(\gamma_k)$ can be calculated in time $O(n)$ by using the Buttom-up Procedure described in [10]. It is remarkable that the special MCF problem on series-parallel networks constructed from a tree by adding the source s to all arcs with $tail(a) = s$ and $head(a) = v$ for all vertices v without predecessors of it can be solved in $O(m\,logm)$ steps [23].

Bein et al. [10] also proved an additional stronger result that if Algorithm 4.4.2 fails to solve minimum cost flow problem, then the underlying network is not a series-parallel.

Ruzika et al. [91] (cf. [96]) modified the MCC Algorithm for TTSP network (Algorithm 4.4.2) by combining the MCC Algorithm for general network (Algorithm 3.5.1) to incorporate the time bound T in the solution. They also exploit the property of series parallel network and showed that there does not exist any cycle with negative cost in the residual network which is not necessary for obtaining minimum cost flow in this special class of networks. Also, the modification in the algorithm rejects a path if its cost (time, in fact) exceeds T, the given time horizon, and thus modified algorithm is applicable on dynamic networks. This modified algorithm (Algorithm 4.4.3) solves maximum dynamic flow problem on TTSP network using a temporally repeated flow over time horizon T [91].

Algorithm 4.4.3 MCC Algorithm for TTSP Network [91]

Input: TTSP network $\mathcal{N} = (V, A, u(a), c(a), s, d, T)$ with integer inputs.
Add an extra arc (d, s) with capacity $u(d, s)$ and cost $-(T + 1)$ on $\mathcal{N}$.
Output: Minimum cost circulation f_m with paths γ_k with corresponding cost $c(\gamma_k)$ and flow value $f(\gamma_k)$.
for all $(v, w) \in A$ **do**
$f(v, w) := 0$, $k := 0$
end for
while there exists a path connecting s and d on $\mathcal{N}$ **do**
$k := k + 1$
Find a minimum cost path γ_k and the corresponding cost $c(\gamma_k)$.
Form a circulation C^k with γ_k using arc (d, s).
if $c(\gamma_k) - (T + 1) < 0$ **then**
$f(\gamma_k) := min\left\{u(v, w) : (v, w) \in C^k\right\}$
else
Stop the algorithm.
end if
for all $(v, w) \in C^k$ **do**
$u(v, w) := u(v, w) - f(\gamma_k)$
if $u(v, w) = 0$ then $A := A - \{(v, w)\}$
end if
end for
end while

The termination of Algorithm 4.4.3 follows due to *'if-conditions'* in the algorithm. For the first case, algorithm stops as soon as costs $c(\gamma_k)$ of the minimum cost path γ_k are greater than time horizon T. For the second case, if this T is large enough to allow all necessary minimum cost paths, algorithm terminates when there are no more paths from s to d on residual network. At the termination, algorithm computes total amount of flow, say, $\mathbf{f}_d$ that is sent from s to d within time horizon T.

Theorem 4.4.3 [91] *Algorithm 4.4.3 solves maximum dynamic flow problem for TTSP network $\mathcal{N}$ with integral capacities and transit time on each arc $a \in A$ optimally.*

As we stated already in Chapter 3 that any maximum dynamic flows on any general network computed as a temporally repeated flows by applying the minimum cost circulation algorithm are not necessarily an earliest arrival flows. However, the Algorithm 4.4.3 indeed solves earliest arrival flow problem for two terminal series parallel network. That is, the maximum dynamic flow solution obtained from Algorithm 4.4.3 has earliest arrival property due to the following theorem (Theorem 4.4.4). However, it is not true that every earliest arrival flow on series-parallel network satisfies the temporarily repeated property.

Theorem 4.4.4 [91] *Let $\mathcal{N} = (V, A)$ be TTSP dynamic network. Let $\boldsymbol{f}_d$ be an optimal solution of MDF problem obtained by applying Algorithm 4.4.3 on $\mathcal{N}$. Then, $\boldsymbol{f}_d$ is also an optimal solution to EAF problem on $\mathcal{N}$.*

Algorithm 4.4.4 DT-EACF Algorithm [31]

1. Given a dynamic TTSP network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ as in Algorithm 4.4.1.
3. Compute dynamic, temporally repeated flow on network $\tilde{\mathcal{N}}$, using Algorithm 4.4.3.
4. Perform the flow decomposition into path and cycle flows of the maximum flow obtained from step-3 and remove all cycle flows.
5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
6. Obtain DT-EACF solution on TTSP network $\mathcal{N}$.

Now consider discrete time earliest arrival contraflow (DT-EACF) problem modeled for TTSP network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$, for T being discretized. The solution procedure has been proposed to DT-EACF problem where arc reversal on the network is allowed only once at time zero in [31]. The procedure (Algorithm 4.4.4) modifies the DT-MDCF

Algorithm 4.4.1 for general networks applying MCC Algorithm for TTSP network (cf. Algorithm 4.4.3) in Step 3. As stated in Subsection 3.5.1, the MCC problem is solved on $\mathcal{N}$ by considering the transit time $\tau(a)$ as cost $c(a)$ for each arc $a \in A$.

We see that due to Theorem 4.4.3, the Algorithm 4.4.4 gives optimal solution to a DT-MDCF problem on TTSP network $\mathcal{N}$. Again, by Theorem 4.4.4, this optimal solution is also an optimal solution to the DT-EACF problem on $\mathcal{N}$. Since all the algorithms applied in Algorithm 4.4.4 run in strongly polynomial time, the following theorem can be stated.

Theorem 4.4.5 [31] *Let $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ be a dynamic TTSP network with integer inputs. Then, Algorithm 4.4.4 solves the DT-EACF problem on $\mathcal{N}$ optimally in strongly polynomial time, if arc reversal is allowed only once at time zero.*

4.4.2 Continuous time maximum dynamic contraflow problem

In network contraflow models discussed in Subsection 4.4.1, time is measured in a discrete manner. Discrete time flow models are computationally easier in comparison to continuous time flow models. However, the latter models naturally better reflect real world evacuation scenarios. In continuous time model, flow units can enter the network at any moment of time if it reaches to the destination before the pre-specified time horizon. Here, we consider maximum dynamic contraflow problem with continuous time setting and call it *continuous time maximum dynamic contraflow (CT-MDCF) problem*. Model introduction and an efficient solution procedure for CT-MDCF problem have appeared in [64].

Continuous time maximum dynamic flow problems in network with zero transit time and time-varying transit time, and storage of flow units at intermediate nodes for later transshipment have been studied in [4] and [84]. The concept of cuts with discrete time setting has also been extended to the case of continuous time setting and established a MaxFlow-MinCut theorem for continuous time setting. This result was later extended to arbitrary transit time on the arcs in [85]. The concept of measure theory can be deployed to design a single model to capture both discrete and continuous aspects of flow problems [71] (cf. [55]). In this model, flow on each arc is modeled as a measure on real line (time axis) which assigns to each suitable subset a real value, interpreted as the amount of flow entering the arc over the subset. They analyzed maximum dynamic flow problem theoretically by introducing the notion of Borel flow. However, we adopt the notion of natural transformation [40] (see

Subsection 3.5.1) for proposing the solution techniques of the continuous time flow problems.

Now, we discuss solution procedure (Algorithm 4.4.5) for CT-MDCF problem for network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs, if arc reversal on $\mathcal{N}$ is allowed only once at time zero. For the solution, we modify Algorithm 4.4.1 by combining it with the concept of natural transformation. It is to be noted that as in case of DT-MDCF problem, if we choose to flip an arc, it remains flipped throughout time horizon. In fact, in this case every continuous flow over time problem can be formulated and solved as a discrete flow over time problem. However, this approach do not remain true for the more general setting where network parameters are subject to fluctuate over time.

Algorithm 4.4.5 CT-MDCF Algorithm [64]

---

1. Given a dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ as in Algorithm 4.4.1.
3. Compute dynamic, temporally repeated flow on network $\tilde{\mathcal{N}}$ for time horizon $T - 1$.
4. Transform discrete dynamic flow into continuous dynamic flow using natural transformation: $f_d(a, t) := f_c(a, [t, t + 1))$ for all $t \in \{0, 1, \ldots, T - 1\}$, [40].
5. Perform flow decomposition into path and cycle flows of maximum flow obtained from step-4 and remove all cycle flows.
6. Arc $(w, v) \in A$ is reversed if and only if flow along arc $(v, w) \in A$ is greater than $u(a)$ or if there is non-negative flow along arc $a \notin A$.
7. Obtain CT-MDCF solution on $\mathcal{N}$ for time horizon T.

There exist temporally repeated flows on a dynamic network with continuous time setting also. Moreover, the flow is maximal due to following lemma.

Lemma 4.4.2 [5] *The temporally repeated flow with continuous time setting is maximal over the time horizon.*

The Lemma 4.4.1 that relies on discrete time setting can be extended to the case for continuous time setting also which states the following:

Lemma 4.4.3 [64] *The maximum flow for two terminal case of maximum dynamic contraflow problem with continuous time setting on $\mathcal{N}$ does not exceed the optimal flow for the corresponding time expanded network $\mathcal{N}^T$.*

Proof: Consider dynamic network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs. Every feasible flow on $\mathcal{N}$ is equivalent feasible flow of the maximum dynamic contraflow problem with continuous time setting on the corresponding time expanded network $\mathcal{N}^T$. Furthermore, the continuous time net flow $\mathbf{f}_c$ does not exceed the discrete time net flow $\mathbf{f}_d$. ■

Algorithm 4.4.5 yields an optimal solution to maximum dynamic contraflow problem with continuous time setting. The following theorem gives proof of correctness of the algorithm.

Theorem 4.4.6 [64] *Let $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ be a dynamic network with integer inputs. Then, Algorithm 4.4.5 yields an optimal solution to CT-MDCF problem on $\mathcal{N}$, if arc reversal is allowed only once at time zero.*

Proof: The CT-MDCF Algorithm is a modified procedure of Algorithm 4.4.1. Steps 2, 3 and 5 are clearly well defined. Flow decomposition breaks the optimal flow into paths from source to sink and into cycles with positive flows. These positive flows vanish in each cycles after cancellation, and ensures that there is either a flow along arc (v, w) or (w, v), but never on both arcs. Hence the resulting flow from step 6 is a feasible flow with arc reversal for $\mathcal{N}$. Step 4 is feasible since the natural transformation converts of a feasible $(T-1)$-horizon maximum dynamic flow in discrete time setting into a feasible T-horizon maximum dynamic flow in continuous time setting.

Since every feasible flow for the continuous time setting on $\tilde{\mathcal{N}}$ is also feasible for the continuous time setting on $\mathcal{N}$, the algorithm is correct for the feasible flow for the continuous time setting.

By the feasibility condition,

$$\tilde{\mathcal{N}}_{CT-MDF_{opt}} \leq N_{CT-MDCF_{opt}}$$

where $\mathcal{N}_{CT-MDCF_{opt}}$ and $\tilde{\mathcal{N}}_{CT-MDF_{opt}}$ stand for the optimal value of the maximum dynamic contraflow on $\mathcal{N}$ and maximum dynamic flow on $\tilde{\mathcal{N}}$, respectively, for continuous time settings.

It is clear by Lemma 4.4.3 that the maximum static contraflow on N^T is not less than the maximum dynamic contraflow on $\mathcal{N}$ in continuous time setting. That is,

$$N_{CT-MDCF_{opt}} \leq N^T_{MSCF_{opt}}$$

where $\mathcal{N}^T_{MSCF_{opt}}$ is the optimal value of the maximum static contraflow on time expanded network $\mathcal{N}^T$.

We have the fact that the maximum static contraflow problem on $\mathcal{N}^T$ is equivalent to the maximum static flow problem on $\tilde{\mathcal{N}}^T$ where the arc set $\tilde{A}$ is defined as $\tilde{A} = \{\tilde{a} := (v,w) : (v,w) \in A \vee (w,v) \in A\}$, capacity $u(\tilde{a}) := u(v,w) + u(w,v)$ and the transit time $\tau(\tilde{a}) := \tau(v,w)$ if $(v,w) \in A$ and $\tau(\tilde{a}) := \tau(w,v)$ otherwise.

Thus,

$$N^T_{MSCF_{opt}} = \tilde{\mathcal{N}}^T_{MSF_{opt}}$$

where $\tilde{\mathcal{N}}^T_{MSF_{opt}}$ stands for the optimal value of the maximum static flow on $\tilde{\mathcal{N}}^T$.

By Lemma 4.4.2, the maximum flow in time expanded network $\tilde{\mathcal{N}}^T$ can be obtained by temporally repeated path flow of a static network $\tilde{\mathcal{N}}$. That is,

$$\tilde{\mathcal{N}}^T_{MSF_{opt}} = \tilde{\mathcal{N}}_{CT-MDF_{opt}}.$$

Hence, we have

$$N_{CT-MDCF_{opt}} \leq N^T_{MSCF_{opt}} = \tilde{\mathcal{N}}^T_{MSF_{opt}} = \tilde{\mathcal{N}}_{CT-MDF_{opt}}.$$

Therefore, $\tilde{\mathcal{N}}_{CT-MDF_{opt}} \leq N_{CT-MDCF_{opt}}$. ■

Theorem 4.4.7 [64] *Algorithm 4.4.5 runs in strongly polynomial time.*

Proof: Finding a temporally repeated flow is equivalent to solving a minimum cost flow problem. The algorithm in [48] for solving this problem leads to a strongly polynomial time of order $O(n^2.m^3.logn)$. Since the natural transformation of $(T-1)$–horizon discrete time maximum dynamic flow yields a T-horizon continuous time maximum dynamic flow, the time complexity of finding a temporally repeated continuous flow is also $O(n^2m^3logn)$. Other significant steps that are required in the algorithm are to solve the maximum static flow problem on $\tilde{\mathcal{N}}$ and to decompose the flow which can be done in $O(n^2m)$ and $O(nm)$ times, respectively. Thus, Algorithm 4.4.5 runs in strongly polynomial time. ■

4.5 Dynamic Contraflow Problems for Multi-network

In the case of unequal to-and-fro transit time on oppositely directed lanes of a road, it is crucial to take network contraflow models with not necessarily equal transit time on anti-parallel arcs into account for preparing evacuation tasks. Moreover, there could be multiple parallel lanes of different transit time on a road. Multi-network flow model could be a tool to incorporate these features in contraflow evacuation models. Here, we allow parallel arcs of different transit time only and anti-parallel arcs with not necessarily equal transit time in multi-network. In this section, we study MDCF problems for multi-network of general type and EACF problems for TTSP multi-network with discrete as well as continuous time settings. Also, we propose efficient solution techniques to them when arc reversibility is allowed only once at time zero like in the case of ordinary networks studied in Section 4.4.

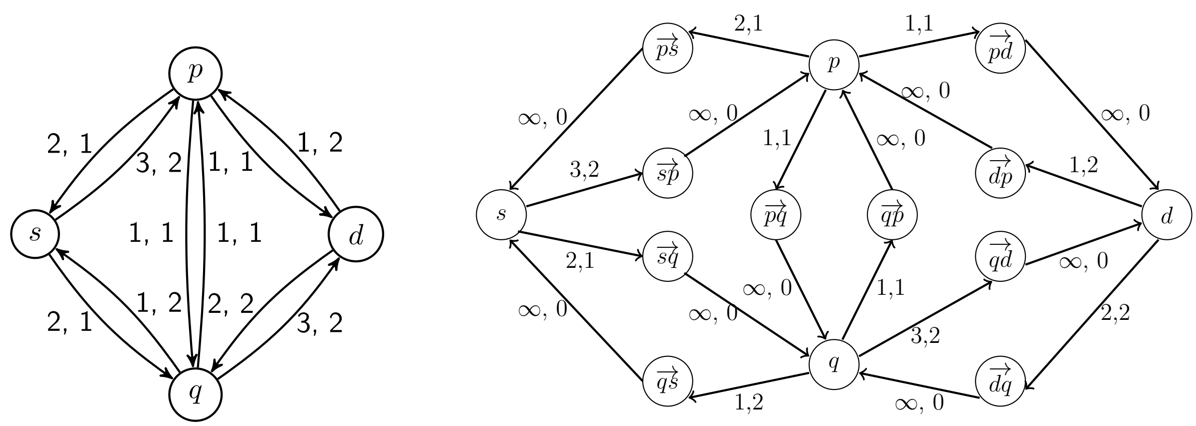


Figure 4.1: A network $\mathcal{N}$ (left) and its corresponding transformed network $\mathcal{N}^+$ with artificial vertices and arcs (right).

We proposed a network modification idea to solve dynamic contraflow problems: namely, maximum dynamic contraflow problem and earliest arrival flow problem, modeled on network with not necessarily equal transit time on anti-parallel arcs, see [12] and [14]. The idea is as follows: Transform input network $\mathcal{N}$ into $\mathcal{N}^+$ by introducing an artificial vertex for each arc to split it into two different arcs: real arc and artificial arc. Assign sufficient capacities and zero transit time to the artificial arcs, whereas the real arcs have capacities and transit time equal to that of the arc before it was split. See Figure 4.1. Then the solution procedures discussed in Section 4.4 are applicable on transformed network $\mathcal{N}^+$. However, in this section, we consider more general model for evacuation problems where parallel as well as anti-parallel arcs on network are allowed and discuss slightly different solution techniques with the details. The model and some results related to these problems have also been included in [16].

4.5.1 MDCF problems for multi-network

Consider a multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$. Here, capacities and transit time on anti-parallel arcs could be unequal. That is, $u(v,w)$ is not necessarily equal to $u(w,v)$, and $\tau(v,w)$ is not necessarily equal to $\tau(w,v)$ for $(v,w),(w,v) \in A$. Moreover, $\tau(v,w)_i$ is not equal to $\tau(v,w)_j$ where $(v,w)_i,(v,w)_j : i \neq j; i,j \in \mathbb{N}$, represent two parallel arcs on $\mathcal{N}$. Let us assume, without loss of generality, that for every arc $(v,w) \in A$ there exists an arc $(w,v) \in A$. One can introduce an artificial arc (w,v) with zero capacity and transit time $\tau(v,w)$, if arc $(w,v) \notin A$ for $(v,w) \in A$. Then maximum dynamic contraflow (MDCF) problem for multi-network $\mathcal{N}$ answers the question: What is the maximum flow from the source s to the sink d on $\mathcal{N}$ within time horizon T, if direction of arcs on $\mathcal{N}$ can be reversed?

While solving MDCF problem over multi-networks, anti-parallel arcs are replaced by a single undirected arc with the same procedure as described in [89], if both arcs have same transit time; and all arcs are kept parallel and undirected with existed arc capacities and transit time, if different transit time exist on them. This generates an undirected multi-network. A multi-network is a network (directed or undirected) in which more than one parallel arcs (directed or undirected) exist between two vertices. The case of an undirected network or a multi-network doesn't differ conceptually from the solution technique point of view for finding maximum flow. In the case of an undirected network $\mathcal{N}$, every undirected arc on $\mathcal{N}$ is replaced by two oppositely directed arcs, each having capacity and transit time equal to that of the old arc. Since there are multiple parallel arcs in a multi-network, one should be careful in handling the arcs. There are no other obstruction with these networks. Parallel arcs (v,w) are labeled as $(v,w)_i$ such that $\tau(v,w)_i < \tau(v,w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$ to avoid obstruction on multi-networks while applying network flow algorithms.

4.5.1.1 Solution for DT-MDCF problem for multi-network

Consider discrete time maximum dynamic contraflow (DT-MDCF) problem on multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$, for T being discretized. The procedure for solving DT-MDCF problem on $\mathcal{N}$, if arc reversibility is permitted only once at time zero, has two phases. At the first, input network $\mathcal{N}$ is transformed into its auxiliary network $\tilde{\mathcal{N}} =$

$(V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ where $\tilde{A}$ contains undirected arc (v, w), if $(v, w), (w, v)$ both belong to A such that $\tau(v, w) = \tau(w, v)$ having capacity and transit time $u(\widetilde{v, w}) := u(v, w) + u(w, v)$ and $\tau(\widetilde{v, w}) := \tau(v, w) = \tau(w, v)$, respectively; and undirected arcs (v, w) and (w, v) both, if (v, w) and (w, v) both belong A such that $\tau(v, w) \neq \tau(w, v)$ without changing capacities and the transit time. It is not known in advance that which of the arcs (v, w) or (w, v) is to be flipped until maximum flow is computed on the reduced network. At the second, a maximum dynamic $s - d$ flow is computed on undirected multi-network $\tilde{\mathcal{N}}$ by using any known algorithm. The procedure for solving DT-MDCF problem on multi-network $\mathcal{N}$ has been summarized in Algorithm 4.5.1.

Algorithm 4.5.1 DT-MDCF Algorithm for Multi-network [16]

1. Given a dynamic multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ in to undirected multi-network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ where
   $\tilde{a} = (v, w) \in \tilde{A}$, if $(v, w), (w, v) \in A$ such that $\tau(v, w) = \tau(w, v)$, with $u(\tilde{a}) = u(v, w) + u(w, v)$ and $\tau(\tilde{a}) = \tau(v, w)$; and
   $\tilde{a} = (v, w) \in \tilde{A}$, if $(v, w) \in A$ and $(w, v) \notin A$ such that $\tau(v, w) = \tau(w, v)$, with $u(\tilde{a}) = u(v, w)$ and $\tau(\tilde{a}) = \tau(v, w)$.
3. Label each parallel arcs $(v, w) \in \tilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$.
4. Compute dynamic, temporally repeated flow on $\tilde{\mathcal{N}}$.
5. Perform the flow decomposition into path and cycle flows of the maximum flow obtained from step-4 and remove all cycle flows.
6. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
7. Obtain DT-MDCF solution on multi-network $\mathcal{N}$.

In the procedure of Algorithm 4.5.1, arc $(w, v) \in A$ is reversed if flow along arc (v, w) exceeds $u(v, w)$ for $\tau(v, w) \leq \tau(w, v)$; or, if, disregard of flow value on (v, w), $\tau(v, w) > \tau(w, v)$. This can be viewed, alternatively, as follows: For $(v, w), (w, v) \in A$ such that $\tau(v, w) = \tau(w, v)$, the flow value at arc $\tilde{a} = (v, w) \in \tilde{A}$ greater than the capacity $u(v, w)$ of the corresponding arc $(v, w) \in A$ means there is flipping of the direction of arc $(w, v) \in A$. Similarly, in the case of unequal transit time, we realize the sense of flipping the direction of arc $a = (w, v) \in A$, if there is some positive flow on corresponding arc $\tilde{a} = (v, w)$. The minimum cost flow algorithm applied in step 3 of Algorithm 4.5.1 to find minimum cost $s - d$ paths ensures that there is flow along arc (v, w) with less or equal transit time in comparison to transit time of corresponding arc (w, v), regardless of the arc (w, v) is saturated.

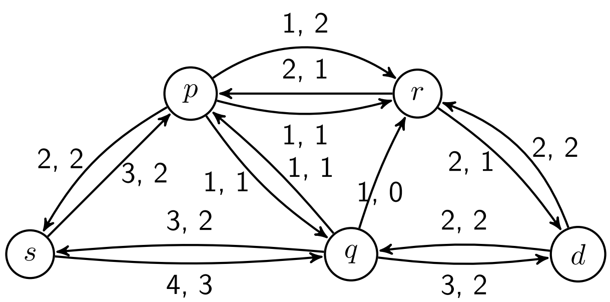


Figure 4.2: Multi-network $\mathcal{N}$ with capacity and transit time next to each arcs.

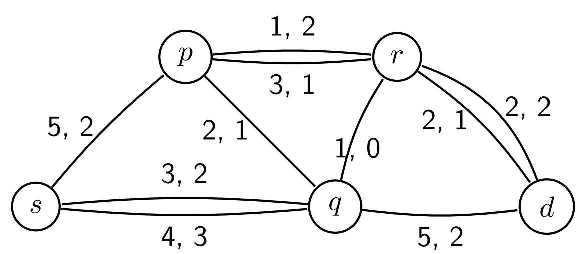


Figure 4.3: Undirected multi-network $\tilde{\mathcal{N}}$ reduced from network $\mathcal{N}$ in Figure 4.2.

Example 4.5.1 *Consider the multi-network given in Figure 4.2 with s the source and d the sink. Applying step 2 of Algorithm 4.5.1 on* $\mathcal{N}$*, we get undirected multi-network* $\tilde{\mathcal{N}}$ *as depicted in Figure 4.3. Before computed dynamic temporally repeated flow on* $\tilde{\mathcal{N}}$*, each undirected arcs is replaced by two oppositely directed arcs with arc capacities and transit time for both arcs equal to that of previous arc. This allows for a flow unit to travel in either direction. However, corresponding arc with opposite direction is deleted from network once the direction of a arc is fixed. The network in its further modified form has been depicted in Figure 4.4. Moreover, the parallel arcs connecting vertices s and p, vertices p and r and vertices r and d are labeled with suffices 1 and 2: suffix 1 for the arc with lesser transit time and suffix 2 for the next arc, for each pair of parallel arcs. Here, flow that reaches at sink d on* $\tilde{\mathcal{N}}$ *for the first time at* $t = 3$ *along path* $s - q - r - d$ *is 1 unit. At time* $t = 4$*, flow of values 3 and 2 can reach the sink along paths* $s - q - d$ *and* $s - p - r - d$*, respectively. On the other hand, no flow reaches at sink d on the original network* $\mathcal{N}$ *at time* $t = 3$ *but reaches at time* $t = 4$ *for the first time with value 1 along each of paths* $s - p - r - d$ *and* $s - p - q - r - d$*. Here, the value of maximum dynamic contraflow on* $\mathcal{N}$ *for time horizon* $T = 10$ *is 59, whereas its value without contraflow for the same time horizon is only 32.*

Now we claim in the following theorem that Algorithm 4.5.1 solves discrete time maximum dynamic contraflow problem on multi-network $\mathcal{N}$ optimally, if arc reversal is allowed only once at time zero.

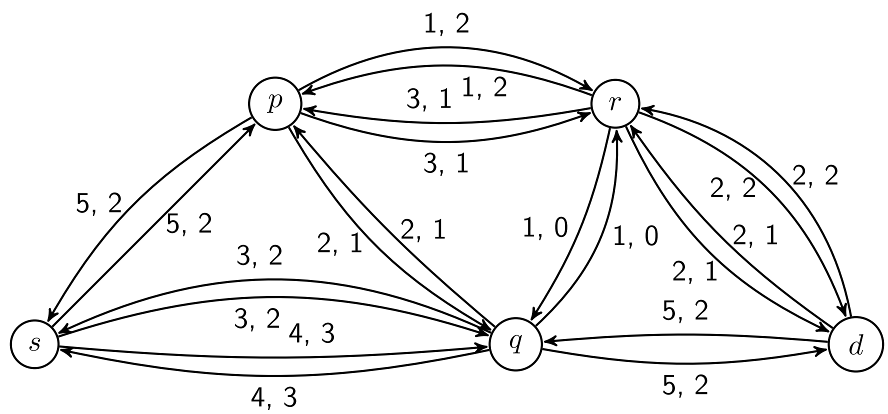


Figure 4.4: Further transformation of the undirected network $\tilde{\mathcal{N}}$ depicted in Figure 4.3 for the purpose of maximum flow computation.

Theorem 4.5.1 [16] *Let $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ be a multi-network. A maximum dynamic flow computed by Algorithm 4.5.1 on $\tilde{\mathcal{N}}$ is equivalent to a maximum dynamic contraflow on $\mathcal{N}$.*

Proof: The auxiliary network $\tilde{\mathcal{N}}$ of original network $\mathcal{N}$ obtained in step 2 of Algorithm 4.5.1 is an undirected multi-network. Now, discrete time maximum dynamic contraflow problem on $\mathcal{N}$ can be viewed as a discrete time maximum dynamic flow problem on $\tilde{\mathcal{N}}$. While solving the latter problem on $\tilde{\mathcal{N}}$, network is to be further transformed by replacing each undirected arc by two oppositely directed arcs with capacities and transit time of both arcs equal to that of original arc. This allows us to send flow on either direction of the arc. However, the flow direction, once chosen, remains fixed throughout the procedure. That is, there is only a flow in one direction of any arc, and never in both directions at the same time as well as at different time periods due to cancellation of flows along the cycles. However, there could be a flow along arc (v, w) and (w, v) such that $\tau(v, w) \neq \tau(w, v)$ for $(v, w), (w, v) \in A$ at the same time or at different time periods. The latter situation does not make the flow in $\tilde{\mathcal{N}}$ an infeasible since, in fact, arcs (v, w) and (w, v) are physically different arcs for $\tau(v, w) \neq \tau(w, v)$, due to the labeling of arcs in step 3. Thus, the flow constructed by Algorithm 4.5.1 is feasible.

Since every feasible flow of maximum dynamic flow problem in the transformed network $\tilde{\mathcal{N}}$ is feasible to maximum dynamic contraflow problem on network $\mathcal{N}$, optimal dynamic flow on $\tilde{\mathcal{N}}$ is not greater than optimal dynamic contraflow on $\mathcal{N}$. On the other hand, due to Lemma 4.4.1, the optimal dynamic contraflow on $\mathcal{N}$ is not greater than optimal static contraflow on time expanded network $\mathcal{N}^T$. By Lemma 4.3.1, this static contraflow is equivalent to optimal static flow on $\tilde{\mathcal{N}}^T$. Again, by Theorem 4.4.1, optimal static flow in $\tilde{\mathcal{N}}^T$ is equivalent to temporally repeated path flow on a static network $\tilde{\mathcal{N}}$. Thus, optimal dynamic contraflow on $\mathcal{N}$ is not greater than optimal dynamic flow on $\tilde{\mathcal{N}}$. ■

Theorem 4.5.2 [16] *Algorithm 4.5.1 runs in strongly polynomial time.*

Proof: Construction of auxiliary network requires only linear time on m. Computing a maximum dynamic flow in step 4 dominates running time of the algorithm. A maximum dynamic flow is computed with the help of temporally repeated flow on $\tilde{\mathcal{N}}$. Finding a temporally repeated flow is equivalent to solving a minimum cost flow problem. The minimum mean cycle-canceling algorithm in [48], for instance, requires $O(n^2m^3logn)$ times for solving this problem. Next effort is to decompose the computed flow which requires $O(mn)$ times [2]. Thus, Algorithm 4.5.1 runs in a strongly polynomial time. ■

4.5.1.2 Solution for CT-MDCF problem for multi-network

Consider the continuous time maximum dynamic contraflow (CT-MDCF) problem for dynamic multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$. Here, we assume that each input parameters are integer and invariant of time. Then the solution algorithm that solves DT-MDCF problem for multi-network can be modified to solve CT-MDCF problem for multi-network. Algorithm 4.5.2 summarizes solution procedure for the problem when arc reversibility is allowed only once at time zero.

Optimality of the solution obtained by Algorithm 4.5.2 can be checked like in the case of DT-MDCF problem for multi-network. Only the additional effort in Algorithm 4.5.2 is to apply notion of transformation of discrete dynamic flow into continuous dynamic flow that always yields a feasible flow [40]. Also, time complexity of finding a temporally repeated continuous flow is equal to time complexity of finding a temporally repeated discrete flow. Therefore, due to Lemmas 4.4.2 and 4.4.3 the following theorem holds true:

Theorem 4.5.3 [16] *Let $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ be a dynamic multi-network with integer inputs. Then Algorithm 4.5.2 solves the CT-MDCF problem for $\mathcal{N}$ optimally in strongly polynomial time.*

MDCF problem for multi-network $\mathcal{N}$ coincides with MDCF problem given in Subsection 4.4.1, if there are equal transit time on each pair of anti-parallel arcs on $\mathcal{N}$ and no parallel

Algorithm 4.5.2 CT-MDCF Algorithm for Multi-network [16]

1. Given a dynamic multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ into $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ as in Algorithm 4.5.1.
3. Label each parallel arcs $(v, w) \in \tilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$.
4. Compute dynamic, temporally repeated flow on network $\tilde{\mathcal{N}}$ for time horizon $T - 1$.
5. Transform discrete dynamic flow into continuous dynamic flow using the natural transformation: $f_d(a, t) := f_c(a, [t, t+1))$ for all $t \in \{0, 1, \ldots, T-1\}$, [40].
6. Perform flow decomposition into path and cycle flows of maximum flow obtained from step-4 and remove all cycle flows.
7. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
8. Obtain CT-MDCF solution for multi-network $\mathcal{N}$ for time horizon T.

---

arcs are allowed on it. And, the contraflow problem on static network coincides with MSCF problem given in Section 4.3.

4.5.2 EACF problems for multi-network

Consider discrete time earliest arrival contraflow (DT-EACF) problem (cf. Subsection 4.4.1) on a TTSP multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs. DT-EACF problem on $\mathcal{N}$ maximizes a feasible net discrete dynamic flow from the source s to the sink d at all time steps $t \in \mathcal{T}$, if direction of arcs on $\mathcal{N}$ can be reversed. Now, we propose a solution procedure (Algorithm 4.5.3) for DT-EACF problem for multi-network $\mathcal{N}$ in summarized form, if direction of arcs are allowed to reverse only once at time zero.

The following straight forward statement is useful in the optimality proof of Algorithm 4.5.3.

Claim 4.5.1 [16] *Two terminal series parallel multi-network $\mathcal{N}$, after transforming into its auxiliary network $\tilde{\mathcal{N}}$, remains two terminal series parallel multi-network.*

Theorem 4.5.4 [16] *Let $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ be a dynamic TTSP multi-network*

Algorithm 4.5.3 DT-EACF Algorithm for Multi-network [16]

1. Given dynamic TTSP multi-network $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs.
2. Transform $\mathcal{N}$ into undirected multi-network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), s, d, T)$ as in Algorithm 4.5.1.
3. Label each parallel arcs $(v, w) \in \tilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$.
4. Compute dynamic, temporally repeated flow on $\tilde{\mathcal{N}}$, using Algorithm 4.4.3.
5. Perform flow decomposition into path and cycle flows of maximum flow obtained from step-4 and remove all cycle flows.
6. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
7. Obtain DT-EACF solution on TTSP multi-network $\mathcal{N}$.

---

with integer inputs. Then Algorithm 4.5.3 solves DT-EACF problem on $\mathcal{N}$ optimally in strongly polynomial time.

Proof: Due to similar arguments as in the proof of Theorem 4.5.1, the construction of auxiliary network $\tilde{\mathcal{N}}$ of network $\mathcal{N}$ is well defined and the transformed network remains a two terminal series-parallel network due to Claim 4.5.1. The dynamic flow obtained from step 4 is optimal for time horizon T, and $\tilde{\mathcal{N}}$ being a TTSP network, the flow computed on it has earliest arrival property [91].

Construction of auxiliary network requires only a linear time on m. The time complexity does not differ for computation of temporally repeated flows on multi-network and general network. Rest of the steps in Algorithm 4.5.3 are same to the steps in Algorithm 4.4.4. Thus, Algorithm 4.5.3 solves DT-EACF problem on multi-network $\mathcal{N}$ for time horizon T optimally in strongly polynomial time. ■

We already mentioned in Chapter 3 that there is an obvious observation that all earliest arrival flows are maximum dynamic flows but the converse is not necessarily true, see Example 4.5.2.

Example 4.5.2 *Consider the multi-network $\mathcal{N}$ given in Figure 4.2 with s the source, d the sink and time horizon $T = 4$. Then maximum dynamic contraflow of value 5 induced by path* *flows* $\{s-q-d; s-p-r-d\}$ *is not an earliest arrival contraflow on* $\mathcal{N}$ *since it is possible to send 1 unit flow already at time 3 along path* $s-q-r-d$. *On the other hand, the dynamic flow of value 5 induced by the path flows* $\{s-q-r-d; s-q-d; s-p-r-d\}$ *is maximum and also an earliest arrival flow on* $\mathcal{N}$ *for* $T=4$.

We see the effect of TTSP network through the following example.

Example 4.5.3 *Consider the multi-network* $\mathcal{N}$ *given in Figure 4.2 with* s *the source,* d *the sink and time horizon* $T=4$. *Remove arcs* (p,q), (q,p) *and* (q,r) *from* $\mathcal{N}$ *to get a TTSP multi-network. Then maximum dynamic contraflow of value 5 induced by the path flows* $\{s-q-d; s-p-r-d\}$ *for* $T=4$ *is, of course, an earliest arrival contraflow on* $\mathcal{N}$.

Together with the notion of natural transformation, Algorithm 4.5.3 also solves continuous time earliest arrival contraflow (CT-EACF) problem on multi-network $\mathcal{N}$. The CT-EACF problem over $\mathcal{N}$ maximizes a feasible net continuous flow from s to d at all time points $t \in [0,T)$, if direction of arcs can be reversed. Optimality and time complexity of the solution procedure can be checked as in the case of CT-MDCF problem for multi-network. That is, CT-EACF problem for multi-network can be solved in strongly polynomial time, when arc reversal capability is allowed only once at time zero.

CHAPTER 5

EVACUATION PLANNING PROBLEMS WITH CAPACITATED VERTICES

5.1 Introduction

Evacuation planning problems modeled with flow conservation at intermediate vertices (existing model) allow the evacuees to leave the source only if they can reach the sink. It is usually not known in advance how many people are at source and how many people does the road topology immediately after disaster (e.g., earthquakes) allow to send into the sink in specified time period. Thus, evacuators must attempt to send out as many evacuees as possible to safer places, despite the sink, where the evacuees can be provided medical aids or other necessary supports or set priorities among evacuees to send to the sink. The relaxation of flow conservation constraints modifies the existing flow model, and it becomes applicable in modeling the situation at which holding of evacuees at safety terminals is possible. The set of safety terminals consists of sink and intermediate vertices, and these are prioritized and capacitated in most of the cases. The prioritization depends upon the evacuation scenario: the facilities at the shelter, the distance from the source or their storage capacities.

This chapter focuses on evacuation planning problems: models and solution procedures, with capability of intermediate holding of evacuees in prioritized basis. We call a lexicographic order, if a flow problem satisfies the prioritization. In particular, we introduce maximum flow evacuation planning problems and discuss their solution techniques for general network in Section 5.2. We consider a uniform path length (UPL) network for the problem and discuss its solution in Section 5.3. Section 5.4 extends the solution to solve an earliest arrival flow problem modeled for uniform path length two terminal series parallel (UPL-TTSP) network. The quickest flow problem with prioritized terminals has been considered in Section 5.5.

We incorporate contraflow approach in models with capacitated vertices in Section 5.6 and discuss some evacuation problems with their solution techniques. Section 5.7 discusses flow problems with continuous time setting in brief. Models and solutions to these problems are also included in [11], [13], [17], [18], [19] and [66].

5.2 Lexicographic Maximum Flow Problem

Recall the dynamic version of maximum flow evacuation planning problem that attempts to send a maximum number of evacuees from risk zone (source) to the safe destination (sink) within given time horizon. In practical applications, there could exist intermediate shelters that are reachable from the source, apart from the sink. Hence, it is advisable to organize an evacuation such that as many evacuees as possible can also be sent to these spots despite sending into the sink. The intermediate shelters might be constrained to some capacities, which restrict the number of evacuees that can be held at them within given time horizon.

We studied maximum flow evacuation planning problem modeled with relaxed flow conservation flow constraint that allows evacuees to be held at temporary shelters at intermediate vertices in [15], [63] and [65]. We proposed efficient solution procedure, by modifying preflow-push algorithm [47] (cf. Subsection 3.4.1), for static case and pseudo polynomial time solution procedure for dynamic case. For modification, the active vertex is relabeled such that its label is at most the label of the source. This allows the excess flow at the vertex to hold on it. The flow model is designed on network with sufficient vertex capacities, and priorities for intermediate spots have been considered in [15] and [65].

Lexicographic maximum flow problem has been investigated as a variant of the classical maximum flow problem with multiple sources and multiple sinks of given priorities and sufficient sink capacities [77, 78, 80]. The authors showed that this problem can be solved in polynomial time. Hoppe and Tardos [57, 58] studied lexicographic maximum dynamic flows. They developed a polynomial time algorithm based on temporally repeated flows that lexicographically maximizes the flow leaving the terminals of an ordered terminal set consisting of sources and sinks. This is equivalent to lexicographically minimizing the flow entering the sinks in the given order.

Here, we aim to lexicographically maximize the amount of flow entering the vertices in terminal set within given time horizon with respect to the prioritization. The terminal set consists of sink and intermediate vertices. This problem is motivated by the situation encountered in evacuation scenarios: as many evacuees as possible are to be sent to *safety*. However, if sending evacuees to *safety* is not possible within given time horizon, it is desirable to send as many evacuees to prioritized shelters of given capacity, which are located within the evacuation zone. We formulate the problem formally in Subsection 5.2.1 and discuss solution strategies for static and dynamic version in Subsections 5.2.2 and 5.2.3, respectively.

5.2.1 Formulation of lexicographic maximum flow problem

Consider a dynamic network $\mathcal{N} = (G, l(a), u(a), k(v), \tau(a), T)$. We assume that the network $\mathcal{N}$ does not contain parallel arcs nor loops. We assume a terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \ldots, v_k\}$ prioritized from higher to lower priority, i.e., $v_1 \succ \cdots \succ v_k$, to be given, where $v_1 = d$. Recall that the vertex capacity function $k : \mathcal{S} \to \mathbb{N}_0$ delimiting the total number of flow units, which may be held in each of the vertices $v \in \mathcal{S}$. We set $k(d) = \infty$ and $k(v)$ to be finite for all $v \in \mathcal{S} \setminus \{d\}$. In the following, we give the flow model by assuming a time horizon $T \in \mathbb{N}$ to be given and treating time in a discrete manner, i.e., $\mathcal{T} := \{0, 1, \ldots, T\}$.

The nonnegative flow variables $f(a, t)$ defined by $f : A \times \mathcal{T} \to \mathbb{N}_0$ that specify the flow over time in the network $\mathcal{N}$ is the number of flow units entering arc a at time step t. The number of flow units entering arc a at time step t are assumed to be bounded by the capacity of an arc, i.e., $f(a, t)$ satisfies the capacity constraints 3.3 for all $a \in A$ and for all $t \in \mathcal{T}$. Moreover, $f(a, t)$ has to be equal to zero for all $t > T - \tau(a)$ and for all $a \in A$. The excess flow at vertex $v \in V$ at time $t \in \mathcal{T}$ is defined as

$$0 \leq ex_f(v, t) := \sum_{a \in \delta^-(v)} \sum_{\xi=0}^{t-\tau(a)} f(a, \xi) - \sum_{a \in \delta^+(v)} \sum_{\xi=0}^{t} f(a, \xi). \tag{5.1}$$

Further, we need to ensure that

$$ex_f(v, T) \leq k(v) \text{ for all } v \in \mathcal{S}. \tag{5.2}$$

Consequently, the total flow of evacuees leaving the source s equals the total flow of the evacuees held at vertices $v \in \mathcal{S}$ over the time horizon T, i.e.,

$$\sum_{a \in \delta^+(s)} \sum_{\xi=0}^{T} f(a, \xi) - \sum_{a \in \delta^-(s)} \sum_{\xi=0}^{T} f(a, \xi) = \sum_{v \in \mathcal{S}} ex_f(v, T). \tag{5.3}$$

The objective function of the maximum flow evacuation planning problem asks to lexicographically maximize the vector $(ex_f(v_1, T), \ldots, ex_f(v_k, T))^\top$ such that $ex_f(v_i, T) \leq k(v_i)$ for $i = 1, \ldots, k$. We call the flow problem on network $\mathcal{N} = (G, l(a), u(a), k(v), \tau(a), T)$ with this objective as *lexicographic maximum dynamic flow problem.* For T being discretized, we abbreviate the problem as DT-LexMDF.

The maximum flow problem with above objective over static network $\mathcal{N} = (G, l(a), u(a), k(v))$ is called *lexicographic maximum static flow (LexMSF) problem.*

5.2.2 Solution for LexMSF problem

Let $\mathcal{N} = (G, l(a), u(a), k(v))$ be a static network without transit times. Further, let $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $d = v_1 \succ \ldots \succ v_k$ prioritized from higher to lower priority. Moreover, we assume that $l(a) = 0$ for all $a \in A$. The goal is to compute a lexicographic maximum flow in $\mathcal{N}$ satisfying the arc capacities for all arcs and the vertex capacities for all $v \in \mathcal{S}$. This is achieved by iterative maximum flow computations in a transformed network as described in the following. First, the network $\mathcal{N}$ with vertex capacities is transformed into a network $\mathcal{N}'$ without vertex capacities. We introduce an artificial vertex v_i' for each vertex $v_i \in \mathcal{S}$. We call the artificial vertex $v_1' := d'$ the supersink. Then, the vertices v_i and v_i' are connected by an artificial arc (v_i, v_i') with $u(v_i, v_i') = k(v_i)$. Moreover, each vertex v_i' is linked to the supersink d' by introducing an artificial arc (v_i', d') having zero arc capacity. Only the artificial arc (d, d') gets infinite arc capacity. Further, every arc (original and artificial) gets a lower arc capacity of zero, i.e., $l(a) = 0$ for all a. Doing this, the network $\mathcal{N} = (G, l, u, k)$ is transformed into the network $\mathcal{N}' = (G', l, u)$ with $G' = (V', A')$, see Figure 5.1.

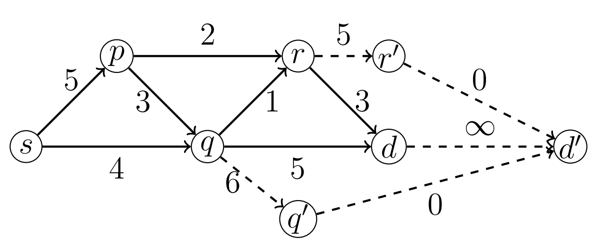


Figure 5.1: Transformed network $\mathcal{N}'$ of the network $\mathcal{N}$ depicted in Figure 3.1 without transit times on arcs and with terminal set $\mathcal{S} = \{d, r, q\}$.

The solution procedure has been given in Algorithm 5.2.1 that works as follows. Since $v_1 = d$ is the vertex with the highest priority, a maximum flow from source s to sink d is computed by applying a maximum flow algorithm in the transformed network $\mathcal{N}'$. Next, lower and upper capacities are updated in the following manner: the lower capacities are set to zero, whereas the upper capacities remain the same; only the arc (d, d') gets a lower and upper capacity equal to the value of the previously computed maximum flow from s to d'. Next, we aim to maximize the flow from s to the vertex v'_i with the next highest priority. This is achieved by setting $u(v'_i, d')$ to infinity and again computing a maximum flow from s to d' with lower and upper arc capacities. Note that we do not have to find a feasible flow in the transformed network, since the maximum flow computed in the previous iteration is already a feasible flow in the modified network. Further, note that due to the lower and upper capacities on (d, d'), it is ensured that the previously computed maximum flow value from s to d' remains the same. This procedure of computing maximum $s - v_i$ flows based on previously computed flows is iteratively repeated for all vertices $v_i \in \mathcal{S}$.

Algorithm 5.2.1 LexMSF Algorithm[19]

Input: A static network $\mathcal{N} = (G, l(a), u(a), k(v))$, $\mathcal{S} = \{v_1, \dots, v_k\} \subset V$ with $d = v_1 \succ \dots \succ v_k$, $l(a) = 0$ for all $a \in A$

Output: A lexicographic maximum static flow satisfying the vertex capacities for all vertices in $\mathcal{S}$

```
d' ←super sink, V ← V ∪ {d'}, A ← A ∪ {(v_1, d')} with u(v_1, d') = ∞
Compute maximum s − d' flow in N and let f* be the optimal value
l(v_1, d') ← f* and u(v_1, d') ← f*
for i = 2, ..., k do
    V ← V ∪ {v'_i}, A ← A ∪ {(v_i, v'_i), (v'_i, d')}
    u(v_i, v'_i) ← k(v_i), l(v_i, v'_i) ← 0
    u(v'_i, d') ← ∞ and l(v'_i, d') ← 0
    Compute maximum s-d' flow in N and let f* be the optimal value
    Let f*(v'_i, d') be the flow value on arc (v'_i, d') w.r.t. f*
    u(v'_i, d') ← f*(v'_i, d') and l(v'_i, d') ← f*(v'_i, d')
```

Example 5.2.1 *Consider the network $\mathcal{N}$ depicted in Figure 3.1 with out transit times on the arcs. Consider the terminal set $\mathcal{S} := \{d, r, q\}$ sorted as $d \succ r \succ q$. Applying Algorithm 5.2.1 on the static network $\mathcal{N}$, we get the maximum flow at the sink d of value 8 and at the intermediate vertices r and q of values 0 and 1, respectively.*

Theorem 5.2.1 [19] *Given a network $\mathcal{N} = (G, l(a), u(a), k(v))$, source s and terminal set $\mathcal{S} = \{v_1, \dots, v_k\} \subset V$ with $d = v_1 \succ \dots \succ v_k$ and $l(a) = 0$ for all $a \in A$. Then, Algorithm 5.2.1 solves the LexMSF problem on $\mathcal{N}$ in strongly polynomial time.*

Proof: Obviously, the maximum flow value f^* from s to d' (see line 2) is equal to the maximum flow value in the original network $\mathcal{N}$ from s to d. Let v_j be the vertex with the next highest priority. One can see that the maximum $s-d'$ flow remains feasible when setting $u(v_1, d')$ and $l(v_1, d')$ to be equal to f^* and introducing arc (v'_j, d') with $u(v'_j, d') = \infty$. If we compute again a maximum $s - d'$ flow, then the flow value on arc (v_1, d') is equal to the maximum s-d'-flow computed in the previous iteration, whereas the flow value on arc (v'_j, d') is equal to the maximal possible flow that can be sent to v'_j among all maximal $s - d'$ flows. Repeating this argument, we get a lexicographical maximum $s - v_i$ flow for all $v_i \in \mathcal{S}$.

The polynomial running time follows from the fact that at most $k < n$ maximum flow problems are to be solved, which can be done in polynomial time. ■

5.2.3 Solution for LexMDF problem

5.2.3.1 LexMDF problem with capacitated vertices

Let $\mathcal{N} = (G, l(a), u(a), \tau(a), k(v), T)$ be a dynamic network. Again, let $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $v_1 \succ \ldots \succ v_k$ be prioritized from higher to lower priority and $l(a) = 0$ for all $a \in A$. We transform the dynamic network $\mathcal{N}$ into a time-expanded network $\mathcal{N}^T = (V^T, A^T, l', u')$ in the following way:

$$V^T := \{v_t \mid t = 0, 1, \ldots, T\},$$
$$A^T := \{(v_t, w_{t'}) \mid (v = w \wedge t' = t + 1) \vee ((v, w) \in A \wedge \tau(v, w) = t' - t)\},$$
$$l'(a) = 0 \text{ for all } a \in A^T,$$
$$u'(v_t, w_{t'}) = u(v, w) \text{ for all } (v, w) \in A \text{ and } u'(v_t, v_{t+1}) = \infty \text{ for all } v \in V.$$

Next, as described in the static version, we introduce a vertex $v'_i := v'_{i_T}$ for all v_{i_T} with $v_i \in \mathcal{S}$. We connect vertices v_{i_T} and v'_i by arcs with upper capacity $k(v_i)$ for all $v_i \in \mathcal{S}$ (and zero transit time, since v_{i_T} and v'_i are on the same time level). Obviously, a static maximum $s_0 - d'$ flow , i.e., a $s_0 - v'_1$ flow, in the time-expanded network corresponds to a discrete dynamic $s - d$ flow in $\mathcal{N}$. After computing the static maximum $s_0 - d'$ flow, we set $l'(d_T, d')$ as well as $u'(d_T, d')$ to the value of the maximum $s_0 - d'$ flow. Afterwards, we introduce an arc from v'_i to d' of infinite capacity, where v_i is the vertex with the next highest priority. Then, we compute again a static maximum $s_0 - d'$ flow in the time-expanded network. This procedure is iteratively repeated (similar to Algorithm 5.2.1) such that we obtain a discrete dynamic lexicographically maximum flow, see Algorithm 5.2.2.

Algorithm 5.2.2 DT-LexMDF Algorithm [19]

Input: A dynamic network $\mathcal{N} = (G, l(a), u(a), \tau(a), k(v), T)$, $\mathcal{S} = \{v_1, \dots, v_k\}$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$

Output: A discrete time lexicographic maximum dynamic flow satisfying the vertex capacities for all vertices in $\mathcal{S}$

Create the time-expanded network as described above
for all $v \in \mathcal{S}$ **do**
  $V^T \leftarrow V^T \cup \{v'\}$, $A^T \leftarrow A^T \cup \{(v_T, v')\}$ ▷ $v' := v'_T$
  $u'(v_T, v') \leftarrow k(v_i)$, $l'(v_T, v') \leftarrow 0$
Compute a static maximum s_0-d'-flow and let f^* be the optimal value
$l'(d_T, d') \leftarrow f^*$ and $u'(d_T, d') \leftarrow f^*$
for $i = 2, \dots, k$ **do**
  $A^T \leftarrow A^T \cup \{(v'_i, d')\}$
  $u'(v'_i, d') \leftarrow \infty$, $l'(v'_i, d') \leftarrow 0$
  Compute a maximum s_0-d'-flow and let f^* be the optimal value
  Let $f^*(v'_i, d')$ be the flow value on arc (v'_i, d') w.r.t. f^*
  $u'(v'_i, d') \leftarrow f^*(v'_i, d')$ and $l'(v'_i, d') \leftarrow f^*(v'_i, d')$

Corollary 5.2.1 [19] *Given a network $\mathcal{N} = (G, l(a), u(a), \tau(a), k(v), T, s, d)$, source s and terminal set $\mathcal{S} = \{v_1, \dots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$ and $l(a) = 0$ for all $a \in A$. Then, Algorithm 5.2.2 solves the DT-LexMDF problem in $\mathcal{N}$ in pseudo-polynomial time.*

Proof: The correctness follows from Theorem 5.2.1. The pseudo-polynomial running time follows from the fact that the size of time-expanded network is pseudo-polynomial in the input size. ■

Example 5.2.2 *Consider the network $\mathcal{N}$ depicted in Figure 3.1 with terminal set $\mathcal{S} := \{d, r, q\}$ sorted as $d \succ r \succ q$ and time horizon $T = 6$. The maximum dynamic flow at the sink d of value 23 is obtained while applying the Algorithm 5.2.2. The algorithm sends flow units of value 3 and 6 at the intermediate vertices r and q which are also optimal for $T = 6$.*

5.2.3.2 LexMDF problem with sufficient vertex capacities

In the following we modify the lexicographic maximum dynamic flow algorithm (cf. [57] and [58]) to solve our problem in the case of sufficient vertex capacities.

Let $\mathcal{N} = (G, u(a), \tau(a), T)$ be a dynamic network with sufficient vertex capacities. Let $\mathcal{S} = \{v_1, \dots, v_k\}$ be a terminal set with $v_1 \succ \dots \succ v_k$ prioritized from higher to lower priority and $l(a) = 0$ for all $a \in A$. We aim to solve our problem in polynomial time without vertex capacities by using the lexicographically maximum dynamic flow algorithm [58]. The algorithm is summarized in Algorithm 5.2.3. Note that $\mathcal{N}^i_{g^{i+1}}$ refers to the residual dynamic network with respect to flow g^{i+1} with vertex set V and arc set A^i.

Algorithm 5.2.3 lexicographically maximizes the amount of flow leaving the terminals in $\mathcal{S}$ in the given order, i.e., it lexicographically maximizes the vector $(-ex_f(v_1, T), \dots, -ex_f(v_k, T)^\top$. However, we aim at lexicographically maximizing the vector $(ex_f(v_1, T), \dots, ex_f(v_k, T)^\top$. Therefore, we adapt our problem in the following way such that we can use Algorithm 5.2.3 to solve it:

1) $V \leftarrow V \cup \{v'_i\}$, $A \leftarrow A \cup \{(v_i, v'_i)\}$ for all $v_i \in \mathcal{S}$
2) $u(v_i, v'_i) \leftarrow \infty$ and $\tau(v_i, v'_i) \leftarrow 0$
3) $\mathcal{S} = \{v'_1, \dots, v'_k, s\}$ with $\mathcal{S}^+ \leftarrow \mathcal{S} \setminus \{s\}$ and $\mathcal{S}^- \leftarrow \{s\}$
4) Take the inverse network of $\mathcal{N}$, i.e., $\mathcal{N}^{inv}$, where all arcs are reversed
5) Apply Algorithm 5.2.3 on $\mathcal{N}^{inv}$

This procedure yields a dynamic flow in $\mathcal{N}^{inv}$ lexicographically maximizing the amount of flow leaving the terminals in $\mathcal{S}$ in the given order. By translating the obtained dynamic flow back to the network $\mathcal{N}$, we obtain the desired dynamic flow lexicographically maximizing the amount of flow entering the terminals in the given order. The correctness of the procedure follows immediately from the correctness of the lexicographically maximum dynamic flow algorithm [58]. Since the overhead of this procedure is determined by the computation of the k minimum cost flows, the polynomial runtime follows.

5.3 Lexicographic Maximum Dynamic Flow Problem on UPL Network

We saw in Section 5.2 that lexicographic maximum dynamic flow (LexMDF) problem with fixed vertex capacities has only the solution that runs in pseudo-polynomial time for general

Algorithm 5.2.3 LexMDF Algorithm for Sufficient Vertex Capacities [58]

Input: A dynamic network $\mathcal{N} = (G, u(a), \tau(a), T)$, $\mathcal{S} = \{v_1, \ldots, v_k\} = \mathcal{S}^+ \dot\cup \mathcal{S}^-$ with $v_1 \succ \cdots \succ v_k$, where $\mathcal{S}^+$ and $\mathcal{S}^-$ refer to the sources and sinks of $\mathcal{S}$, respectively

Output: A lexicographic maximum dynamic flow

$V \leftarrow V \cup \{s^*\}$ ▷ Introduce super source s^*

$A^{k+1} \leftarrow A \cup \{(s^*, s) \mid s \in \mathcal{S}^*\}$, $u(s^*, s) \leftarrow \infty$, $\tau(s^*, s) \leftarrow 0$

$g^{k+1} \leftarrow 0$ ▷ zero flow

$\Gamma^{k+1} \leftarrow \emptyset$ ▷ path decomposition

for $i = k, \ldots, 1$ **do**

    $A^i \leftarrow A^{i+1}$

    if $v_i \in \mathcal{S}^-$ **then**

        $A^i \leftarrow A^i \cup \{(v_i, s^*)\}$, $u(v_i, s^*) \leftarrow \infty$, $\tau(v_i, s^*) \leftarrow -(T+1)$

        $f^i \leftarrow$ min cost circulation in $\mathcal{N}^i_{g^{i+1}}$ with τ as arc costs

    if $v_i \in \mathcal{S}^+$ **then**

        $A^i \leftarrow A^i \setminus \{(s^*, v_i)\}$

        $f^i \leftarrow$ min. cost max. s^*-v_i-flow in $\mathcal{N}^i_{g^{i+1}}$ with τ as arc costs

    $g^i \leftarrow g^{i+1} + f^i$

    $P^i \leftarrow$ path decomposition of f^i

    $\Gamma^i \leftarrow \Gamma^{i+1} \cup P^i$

return Γ^1

network. In this section, we study the problem for discrete time setting modeled in uniform path length (UPL) network and propose a polynomial time solution based on temporally repeated flows. Let $\mathcal{N}$ be any two terminal network with source vertex s. Then the network $\mathcal{N}$ is said to be a uniform path length (UPL) network if, for any vertex $v \in \mathcal{N}$, all possible directed $s - v$ path on $\mathcal{N}$ have equal distances. We consider the distance of the path with respect to its transit time. That is, a network $\mathcal{N}$ is a uniform path length network for which the sum of the transit times on arcs on any possible path from the source s to any vertex $v \in \mathcal{N}$ are equal. A UPL network with respect to transit time has been depicted in Figure 5.2.

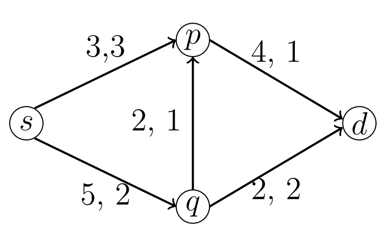


Figure 5.2: A uniform path length (UPL) network $\mathcal{N}$ with source vertex s.

5.3.1 Solution to LexMDF problem on UPL network

Consider a uniform path length (UPL) network $\mathcal{N} = (G, u(a), k(v), \tau(a), s, d, T)$ where the terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \ldots, v_k\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_k$, to be given. Here, the goal is to solve the LexMDF problem on $\mathcal{N}$ in polynomial time using temporally repeated flows. The main idea of the solution procedure of the maximum dynamic flow problem on $\mathcal{N}$ is to find the $s - v_i$ paths at all possible time steps $t \in \mathcal{T}$ with corresponding flow value and send as many units of flow as possible along paths as long as possible. Such paths can be found by decomposing the flow on solving *Lexicographic Minimum Cost Circulation (LexMCC) problem* on $\mathcal{N}$ iteratively.

The minimum cost circulation algorithm (Algorithm 3.5.1), for example, can be applied to solve LexMCC problem on $\mathcal{N}$ repeatedly for each $v_i \in \mathcal{S}$ as a sink in given priority order on corresponding residual network of $\mathcal{N}$ with additional arc (v_i, s) with capacity equal to $k(v_i)$ and transit time $-(T+1)$. Also, the transit time $\tau(a)$ for all $a \in A$ is switched into the cost $c(a)$. This yields a set Γ_{v_i} of all $s - v_i$ paths, denoted as Γ_{v_i}, that could be temporally repeated from time step zero for each $v_i \in \mathcal{S}$. It is noteworthy to mention that path γ_{v_i} is a chain of vertices and arcs in the network $\mathcal{N}$ starting at the source s and terminating at vertex v_i. To each path γ_{v_i}, we associate the following information: (a) $f(\gamma_{v_i})$ – the flow value that can be sent along γ_{v_i} at once, (b) $\tau(\gamma_{v_i})$ – the time required to travel γ_{v_i} by a flow unit, (c) $I_t(\gamma_{v_i})$ – the time step at which the flow along γ_{v_i} starts to get repeated and (d) $F_t(\gamma_{v_i})$ – the time step after which the flow along γ_{v_i} stops to get repeated. The procedure of solving LexMCC problem is termed as LexMCC Algorithm hereafter.

Lemma 5.3.1 [17] *Given a UPL network $\mathcal{N}$ with prioritized set of vertices $\mathcal{S} \subset V$. Then LexMCC problem can be solved in $O(n \times MCF(n,m))$ times on $\mathcal{N}$ where $MCF(n,m)$ is the time complexity for single MCF problem.*

Proof: Lemma follows directly from the fact that $|\mathcal{S}| < |V| = n$. ■

Temporally repeated flows (TRFs) generate feasible optimal flow for $v_1 = d$ as sink for the problem [42]. However, while considering remaining vertices $v_i \in \mathcal{S}$ as the sink, flow computed by TRFs may exceed fixed vertex capacities. Moreover, TRFs may not induce optimal flows for these vertices as sinks due to non-uniqueness of path decomposition carried out for TRFs. These hurdles occur due to the fixed vertex capacities at intermediate vertices. In the following we fix the hurdle associated to TRFs.

5.3.1.1 Construction of extended set $\Gamma^E_{v_i}$

Consider a UPL network $\mathcal{N} = (G, u(a), k(v), \tau(a), s, d, T)$. Any temporally repeated flow on $\mathcal{N}$ generated by Γ_{v_i}, the set of $s - v_i$ paths obtained by applying LexMCC algorithm, has limitation. The limitation is that there may exist $s - v_i$ path, say γ_{v_i} such that $\gamma_{v_i} \notin \Gamma_{v_i}$, for $v_i \in \mathcal{S}$, on the residual network of $\mathcal{N}$ for an interval of time with transit time $\tau(\gamma_{v_i}) < T + 1 - I_t(\gamma_{v_i})$ along which some flow units could be sent at v_i. This situation occurs when any path $\gamma_{v_j} \in \bigcup_{j=1}^{i-1} \Gamma_{v_j}$ is free to carry flow units at v_i at time $I_t(\gamma_{v_i}) > 0$, due to the time limit or vertex capacity at vertex v_j for some $j < i$, before the time $T + 1 - \tau_{\gamma_{v_i}}$. In this situation, $I_t(\gamma_{v_i}) = F_t(\gamma_{v_j}) + 1 - N(\gamma_{v_j})$ where $N(\gamma_{v_j})$ is the actual number of times that the flow along γ_{v_j} is repeated. The number of actual repetitions $N(\gamma_{v_i})$ along any path v_i depends upon the vertex capacity $k(v_i)$ and is given by the Path Flows Repetition (PFR) technique discussed in next sub-topic. Thus, applying lexMCC Algorithm at time zero only may not be enough for the optimal solution at all possible vertices using the TRF approach. And, it is required to find an extended set $\Gamma^E_{v_i}$ that contains all possible $s - v_i$ paths, say γ_{v_i}, which could be started to repeat at time $I_t(\gamma_{v_i}) \geq 0$.

An extended set of paths $\Gamma^E_{v_i}$ is given by

$$\Gamma^E_{v_i} := \begin{cases} \Gamma_{v_i} & \text{for } i = 1 \\ \Gamma_{v_i} \cup \Gamma'_{v_i} & \text{for } i > 1 \end{cases}$$

where Γ'_{v_i} is the set of all $s - v_i$ paths that are free to carry flow units at v_i at time intervals $I_1(\gamma_{v_{i-1}}) = [I_t(\gamma_{v_{i-1}}), F_t(\gamma_{v_{i-1}}) - N(\gamma_{v_{i-1}})]$ and $I_2(\gamma_{v_{i-1}}) = [F_t(\gamma_{v_{i-1}}) + 1, T]$ with respect to each path $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$. These two intervals are the complement of the interval of time period in which the path $\gamma_{v_{i-1}}$ is engaged in sending flow units at vertex v_{i-1}, given by $[F_t(\gamma_{v_{i-1}}) + 1 - N(\gamma_{v_{i-1}}), F_t(\gamma_{v_{i-1}})]$, on the time horizon (interval) $[I_t(\gamma_{v_{i-1}}), T]$. The first interval $I_1(\gamma_{v_{i-1}})$ is discarded if $F_t(\gamma_{v_{i-1}}) - N(\gamma_{v_{i-1}}) < I_t(\gamma_{v_{i-1}})$, and the second interval $I_2(\gamma_{v_{i-1}})$ is discarded if its own immediate parent interval is $I_1(\gamma_{v_{i-2}})$. If no interval is discarded, they are merged in a single interval $[I_t(\gamma_{v_{i-1}}), T]$ if $N(\gamma_{v_{i-1}}) = 0$, and taken as two different intervals if $N(\gamma_{v_{i-1}}) > 0$. It is to be noted that $I_1(\gamma_{v_1}) = \emptyset \; \forall \; \gamma_{v_1} \in \Gamma_{v_1}$.

Network $\mathcal{N}_{\Gamma_{v_k}}$, the residual network of $\mathcal{N}$ after solving LexMCC problem on $\mathcal{N}$, is renovated with respect to the path $\gamma_{v_{i-1}}$ for corresponding free time intervals I_1 and I_2 separately. The renovation is crucial not to lose the availability of network portion at any time steps within given time horizon. Then LexMCC Algorithm is applied on the renovated network to find the set Γ'_{v_i}. During the renovation of $\mathcal{N}_{\Gamma_{v_k}}$ with respect to path $\gamma_{v_{i-1}}$, the capacity of each arc $a \in \mathcal{N}_{\Gamma_{v_k}}$ is increased by $f(\gamma_{v_{i-1}})$ if the arc $a = (v, w)$ also belongs to path $\gamma_{v_{i-1}}$, and the capacity of the arc $(w, v) \in \mathcal{N}_{\Gamma_{v_k}}$ is decreased by the same value $f(\gamma_{v_{i-1}})$. That is,

$$u(a) := \begin{cases} u(a) + f(\gamma_{v_{i-1}}) \text{ for } a = (v, w) \in \mathcal{N}_{\Gamma_{v_k}} \text{ such that } a = (v, w) \in \gamma_{v_{i-1}} \\ u(a) - f(\gamma_{v_{i-1}}) \text{ for } a = (w, v) \in \mathcal{N}_{\Gamma_{v_k}} \text{ such that } a = (v, w) \in \gamma_{v_{i-1}}. \end{cases}$$

The LexMCC Algorithm is applied only after renovation of residual network $\mathcal{N}_{\Gamma_{v_k}}$ with respect to all paths $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$ that are free at the same interval of time. This significantly reduces computational complexity (application of MCC Algorithm) of the entire algorithm. This is not possible if $\mathcal{N}$ is not a UPL network. While choosing the second, third and so on paths for renovation, the network which is renovated with respect to first, second and so on paths, respectively, is renovated. This process is repeated for all $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$.

To compute a maximum dynamic flow on $\mathcal{N}$, it is required to repeat the path flows in extended set $\Gamma^E_{v_i}$ by respecting the vertex capacities $k(v_i)$ at vertex v_i for each $v_i \in \mathcal{S}$. For this, the following *Path Flows Repetition (PFR) Technique* is proposed.

5.3.1.2 Path flows repetition (PFR) technique

Paths $\gamma_{v_i} \in \Gamma^E_{v_i}$ are indexed as $\gamma^p_{v_i}, p = 1, 2, ..., l$, in such a way that the path with highest final time, $F_t(\gamma^p_{v_i})$ among the paths $\gamma_{v_i} \in \Gamma^E_{v_i}$ with highest initial time, $I_t(\gamma^p_{v_i})$ gets the least vertex exponent p and so on. If two paths $\gamma_{v'_i}$ and $\gamma_{v''_i}$ have same final time, choice of path depends on the priority vertex the paths pass through. For example, if path $\gamma_{v'_i}$ passes through the vertex v_{i-1} and the path $\gamma_{v''_i}$ passes through the vertex v_{i-2} whiling reaching at vertex v_i, we choose the path $\gamma_{v''_i}$. Tie after this can be broken arbitrarily.

The computation of TRF $\mathbf{f}(\gamma^p_{v_i}) := \sum_{q=1}^{p}(T + 1 - I_t(\gamma^q_{v_i}) - \tau(\gamma^q_{v_i}))f(\gamma^q_{v_i})$ for vertex v_i starts with $p = 1$. If $\mathbf{f}(\gamma^p_{v_i}) = k(v_i)$, $\mathbf{f}(\gamma^p_{v_i})$ is a maximum flow for v_i. If $\mathbf{f}(\gamma^p_{v_i}) < k(v_i)$, $\mathbf{f}(\gamma^{p+1}_{v_i})$ is computed if $p + 1 \leq l$, otherwise $\mathbf{f}(\gamma^p_{v_i})$ is maximum. If $k(v_i) < \mathbf{f}(\gamma^p_{v_i})$, $k(v_i)$ is maximum flow if $p = 1$ and $\mathbf{f}(\gamma^{p-1}_{v_i}) + k_r(v_i)$ is maximum if $p > 1$ at the vertex v_i. The TRF is likely to get flow repeated more than once over the time horizon T. If flow repetition occurs more than once along the path $\gamma^p_{v_i}$ over T, the time interval $T' = \left[I_t(\gamma^p_{v_i}) + \tau_{\gamma^p_{v_i}}, T\right]$ is halved and the TRF is computed in the second half. The computed flow is then added to $\mathbf{f}(\gamma_{v_i}^{p-1})$. The total flow is compared to the vertex capacity. Flow in the first half is also computed and then added if the total flow is less than the vertex capacity. If the total flow exceeds the vertex capacity, the added flow is discarded. Then the second half is further halved and the procedure is repeated. Integral time units of the time horizon T is preserved by rounding up or down to the nearest integer during halving the interval. The procedure is executed either the total flow equals the vertex capacity or $l < p$.

A flow with value more than the residual vertex capacity $k_r(v_i)$ may occur along the path $\gamma_{v_i}^p$ while sending even at once at the vertex v_i. In this situation, the set $\Gamma_{v_i}^E$ is updated by splitting $\gamma_{v_i}^p$ into $\gamma_{v_i}^{p'}$ and $\gamma_{v_i}^{p''}$ with flow values $k_r(v_i)$ and $f(\gamma_{v_i}^p) - k_r(v_i)$, respectively.

Algorithm 5.3.1 summarizes the procedure that yields the maximum flow on network $\mathcal{N}$ at each of the possible vertices in given priority order.

Algorithm 5.3.1 DT-LexMDF Algorithm for UPL Network [17]

1. Given a dynamic UPL network $\mathcal{N} = (G, u(a), \tau(a), k(v), s, d, T)$, $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $d = v_1 \succ \cdots \succ v_k$.
2. Find Γ_{v_i} $\forall\, i = 1, 2, \ldots, k$ by solving the LexMCC problem on $\mathcal{N}$ with additional arcs (v_i, s) with capacity $k(v_i)$ and transit times $-(T+1)$.
3. For $i = 1$, set $\Gamma_{v_i}^E = \Gamma_{v_i}$ and apply PFR technique on $\Gamma_{v_i}^E$. For $i > 1$, go to step 4.
4. For each path $\gamma_{v_{i-1}} \in \Gamma_{v_{i-1}}^E$, find the interval $[F_t(\gamma_{v_{i-1}}) + 1 - N(\gamma_{v_{i-1}}), F_t(\gamma_{v_{i-1}})]$ and intervals $I_1 = [I_t(\gamma_{v_{i-1}}), F_t(\gamma_{v_{i-1}}) - N(\gamma_{v_{i-1}})]$ and $I_2 = [F_t(\gamma_{v_{i-1}}) + 1, T]$.
5. Renovate the network $\mathcal{N}_{\Gamma_{v_k}}$ with respect to path $\gamma_{v_{i-1}}$ for intervals I_1 and I_2.
6. Find Γ'_{v_i} $\forall\, i = 2, \ldots, k$ by solving the LexMCC problem on renovated $N_{\Gamma_{v_k}}$ as initial time $I_t(\gamma_{v_i})$ with additional arcs (v_i, s) with capacity $k(v_i)$ and transit times $-(T+1)$.
7. Set $\Gamma_{v_i}^E = \Gamma_{v_i}$ and update $\Gamma_{v_i}^E := \Gamma_{v_i}^E \cup \Gamma'_{v_i}$ $\forall\, i = 2, \ldots, k$.
8. Apply PFR technique on $\Gamma_{v_i}^E$.
9. Obtain dynamic $s - v_i$ flow on $\mathcal{N}$.

Lemma 5.3.2 [17] *In Algorithm 5.3.1, LexMCC problem is solved at most* $2n$ *times for each* $v_i \in \mathcal{S}$.

Proof: At first we prove that the application of PFR technique on $\Gamma^E_{v_i}$, for each $v_i \in \mathcal{S}$, creates at most 2 new free time intervals for next prioritized vertex v_{i+1}. For, let TRF $\mathbf{f}_{\gamma^p_{v_i}}$ be any optimal flow for v_i on $\mathcal{N}$ obtained by the application of PFR on $\Gamma^E_{v_i}$. Also, let $\gamma^p_{v_i}$ is the path that exists in the interval $[I_t(\gamma_{v_i}), F_t(\gamma_{v_i})]$. Here, if TRF $\mathbf{f}_{\gamma^p_{v_i}}$ is obtained when all the paths that exist to carry flow at v_i are repeated temporally for all time steps in the interval, no new free time interval for v_{i+1} is formed. If all of such path are repeated for equal number of times less than the maximum possible time steps in the interval, only one new free time interval is formed. And, if any one of such paths needs to be split in to two paths and repeated one of them for some less or more number of times than other, one extra new free time interval is formed for next prioritized vertex v_{i+1}.

For $i = 1$, the MCC Algorithm is applied only once and twice for $i = 2$; once with initial time as zero and next with initial time as $F_t(\gamma_{v_1})$, being sufficient capacity at $v_1 = d$. However, for $i > 2$, the number of times for the application of the algorithm is increased by at most 2 in each of at most n iterations, due to above argument and since it is applied only after the the renovation of $\mathcal{N}_{\Gamma_{v_k}}$ with respect to all paths $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$ that are free at the same interval of time. Therefore, to compute the extended set $\Gamma^E_{v_i}$, LexMCC problem is solved at most $2n$ times for each $v_i \in \mathcal{S}$. ■

Lemma 5.3.3 [17] *Renovation of the residual network $\mathcal{N}_{\Gamma_{v_k}}$ is well defined for each iteration.*

Proof: The residual network $\mathcal{N}_{\Gamma_{v_k}}$ is well defined by its definition. It is renovated with respect to each path $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$ that exist in the same interval of time by taking any one of such paths at the first. While choosing the second, third and so on paths, the network which is renovated with respect to first, second and so on paths, respectively, is considered for renovation. During renovation with respect to path $\gamma_{v_{i-1}}$, the capacity of each arc $a = (v, w) \in N_{\Gamma_{v_k}}$ is increased by $f(\gamma_{v_{i-1}})$ if the arc a also belongs to $\gamma_{v_{i-1}}$, and the capacity of the arc $(w, v) \in N_{\Gamma_{v_k}}$ is decreased by the same value $f(\gamma_{v_{i-1}})$. The renovation of the network is done only for those interval of time which was never been used by the path $\gamma_{v_{i-1}}$ during temporal repetition. Thus, the renovation of the residual graph $\mathcal{N}_{\Gamma_{v_k}}$ is well defined for each iteration. ■

Lemma 5.3.4 [17] *For any $v_i \in \mathcal{S}$, the number of paths in the extended set $\Gamma^E_{v_i}$ is bounded above by $2nm$.*

Proof: By Lemma 5.3.2, LexMCC problem is solved at most $2n$ times for each $v_i \in \mathcal{S}$. And, at most m minimum cost flow paths from the source s to the vertex v_i do exist in each iteration. Therefore, the number of paths in $\Gamma^E_{v_i}$, for $v_i \in \mathcal{S}$, does not exceed $2nm$. ■

Lemma 5.3.5 [17] *The residual network $\mathcal{N}_{\Gamma_{v_k}}$ is renovated in time $O(nm)$ for each $v_i \in \mathcal{S}$.*

Proof: For each vertex $v_i \in \mathcal{S}$, the residual network $\mathcal{N}_{\Gamma_{v_k}}$ is renovated with respect to each path $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$ separately. By Lemma 5.3.4, there are at most $2nm$ paths in $\Gamma^E_{v_i}$. Therefore, the number of iterations for renovation of network $\mathcal{N}_{\Gamma_{v_k}}$ for a vertex $v_i \in \mathcal{S}$ is $2nm$. This concludes that the residual network $\mathcal{N}_{\Gamma_{v_k}}$ can be renovated in time $O(nm)$ for each $v_i \in \mathcal{S}$. ■

Lemma 5.3.6 [17] *The PFR technique executes in time $O(nm + logT)$.*

Proof: The extended set $\Gamma^E_{v_i}$ has at most $2nm$ paths, by Lemma 5.3.4. Therefore, TRF $\mathbf{f}_{\gamma^p_{v_i}}$ is computed on $\Gamma^E_{v_i}$ and compared to the vertex capacity $k(v_i)$ at most $2nm$ times. Additionally, while the computed TRF $\mathbf{f}_{\gamma^p_{v_i}}$ exceeds the vertex capacity at v_i, the interval $T' = \left[I_t(\gamma^p_{v_i}), F_t(\gamma^p_{v_i})\right]$ is halved and the TRF is computed in one of the half intervals. This process needs repetition until the length of halved interval is unity in worst case. Therefore, this process takes $O(logT)$ time to execute. This concludes that the PFR technique executes in time $O(nm + logT)$. ■

Theorem 5.3.1 [17] *Given a UPL network $\mathcal{N} = (G, u(a), k(a), \tau(a), T)$, source s and terminal set $\mathcal{S} = \{v_1, \ldots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$. Then Algorithm 5.3.1 yields an optimal solution to the LexMDF problem on $\mathcal{N}$.*

Proof: As the vertex $v_1(= d)$ has sufficient storage capacity, applying Path Flows Repetition technique for this vertex as the sink is equivalent to pushing the flow units with value $f(\gamma_{v_1})$ along each path $\gamma_{v_1} \in \Gamma_{v_1}$ for each time step $t \in \left\{0, 1, \ldots, T - \tau_{\gamma_{v_1}}\right\}$ from the source s to the sink d. This temporally repeated flow for sink d induces a dynamic $s - d$ flow which is feasible and optimal [43].

For each $i > 1$, the extended set $\Gamma^E_{v_i}$ contains all minimum cost $s - v_i$ paths that exist at any time step $t \in \{0, 1, \ldots, T\}$ on residual network of $\mathcal{N}$ with respect to the optimal flow

$\mathbf{f}_{\gamma^p_{v_{i-1}}}$ at previous immediate prioritized vertex v_{i-1}. Thus, the TRF $\mathbf{f}_{\gamma^p_{v_i}}$ obtained by applying PFR technique on $\Gamma^E_{v_i}$ is feasible. The technique pushes flows of corresponding values along each path as long as possible unless $k(v_i)$ is satisfied. Moreover, the flow is pushed along the paths in $\Gamma^E_{v_i}$ with the strategy of saving unused paths in $\Gamma^E_{v_i}$ for the use of next less prioritized vertex v_{i+1} without violating the optimality at v_i. This is assured by selecting the path with highest final time, $F_t(\gamma_{v^p_i})$ among the paths $\gamma_{v_i} \in \Gamma^E_{v_i}$ with highest initial time, $I_t(\gamma_{v^p_i})$ at the first and so on. Thus, TRF $\mathbf{f}_{\gamma^p_{v_i}}$ is optimal on $\mathcal{N}$ for each $v_i \in \mathcal{S}$. ■

Theorem 5.3.2 [17] *Algorithm 5.3.1 runs in strongly polynomial time.*

Proof: Due to Lemma 5.3.1 and 5.3.2, the LexMCC problem can be solved in time $O(n \times MCF(n,m))$ for at most $2n$ times for each of at most n vertices $v_i \in \mathcal{S}$. Due to Lemma 5.3.5, the residual network $\mathcal{N}_{\Gamma_{v_k}}$ is renovated in time $O(nm)$ for each $v_i \in \mathcal{S}$. The Path Flows Repetition technique can be performed in $O(nm + logT)$ time for each vertex $v_i \in \mathcal{S}$, by Lemma 5.3.6. If one wishes to apply the MCF algorithm of [48], for example, Algorithm 3.5.1 has complexity of order $O(n^2\, m^3\, logn)$. Thus, Algorithm 5.3.1 runs in $O(n^3\,(n^2\, m^3\, logn) + n(nm) + n(nm + logT))$. Equivalently, the algorithm has time complexity of order $O(n^5\, m^3\, logn + n^2\, m + n\, logT)$ which is strongly polynomial. ■

5.4 Lexicographic Earliest Arrival Flow Problem on UPL-TTSP Network

Despite the importance in evacuation planning, earliest arrival flow does not necessarily exist for a general network with multiple sinks, even if there is a single source [9], [38]. However, the problem on network with multiple sinks where all arc transit time are zero have been studied and proposed polynomial time algorithms in [92] and [61]. The problem in [61] aims to maximize the ratios of flow values to capacities on the sinks lexicographically instead of strictly obeying the capacity constraints on them. A pseudo-polynomial time algorithm for solving the problem with arbitrary transit time for arc has also been proposed.

Here, we study the problem with a different scenario where a set $\mathcal{S}$ of prioritized terminals, as defined in Subsection 5.2.1, is given instead of multiple sinks. For given a network with terminal set $\mathcal{S}$, a discrete time lexicographic maximum dynamic flow problem that fulfills its objective at each time step $t \in \mathcal{T}$ is a *discrete time lexicographic earliest arrival flow (DT-LexEAF) problem.* That is, the objective of a DT-LexEAF evacuation planning problem is to send a maximum number of evacuees at the possible earliest time from risk zone to the safety zone together with relatively safe prioritized intermediate spots within given time horizon T in given priority order. It is clear that every earliest arrival flow is a maximum dynamic flow for given time horizon. However, the converse is not always true for general network. In this section, we purpose an efficient solution procedure that obtains a lexicographic maximum dynamic flow on a typical network with non-zero transit time on arcs, and claim that this flow schedule has an earliest arrival property.

Consider a uniform path length two terminal series parallel (UPL-TTSP) network $\mathcal{N} = (G, u(a), k(a), \tau(a), T)$ with terminal set $\mathcal{S} \subset V$ as in the case of LexMDF problem in Subsection 5.2.1. Here, the vertex v_1 always gets sufficient storage capacity whereas vertices v_i for $i \neq 1$, get either zero or sufficient storage capacities. That is, not all vertices in V have storage capacities on them. With these settings the LexEAF problem on $\mathcal{N}$ aims to maximize the flow units sent to the terminals in $\mathcal{S}$ in given priority order at each time step $t \in \mathcal{T}$.

The solution strategy to DT-LexEAF problem on UPL-TTSP network $\mathcal{N}$ is similar to that of EAF problem on TTSP network given in [91]. The MCC Algorithm 4.4.3 is applied to solve LexMCC problem on $\mathcal{N}$ repeatedly for each $v_i \in \mathcal{S}$ with additional arc (v_i, s) with capacity equal to $k(v_i)$ and transit time $-(T+1)$. This yields a set Γ_{v_i} of all $s - v_i$ paths that could be temporally repeated for each $v_i \in \mathcal{S}$. However, dynamic flow generated by temporally repeated flow along the paths obtained by solving this problem may not be optimal on $\mathcal{N}$ at all possible vertices. This hurdle can be overcome by the construction of extended set of paths $\Gamma^E_{v_i}$ as in the case of DT-LexMDF problem in Section 5.3 for UPL network. Here, $k(v_i)$ being sufficient, there does no exist the free time interval I_1 which significantly reduces computational complexity of the LexMCC problem. The set $\Gamma^E_{v_i}$ induces an optimal dynamic flow for each v_i on $\mathcal{N}$. Exact solution procedure is given in Algorithm 5.4.1.

Theorem 5.4.1 [17] *Algorithm 5.4.1 yields an optimal solution to the DT-LexEAF problem on UPL-TTSP network $\mathcal{N}$ in polynomial time.*

Proof: The algorithm pushes flow of value $f(\gamma_{v_i})$ along each path on $\Gamma^E_{v_i}$ for each possible time step $t \in \{0, 1, 2, ..., T - \tau_{\gamma_{v_i}}\}$ from the source vertex s to each of the destination vertex

Algorithm 5.4.1 DT-LexEAF Algorithm for UPL-TTSP Network [17]

1. Given a UPL-TTSP network $\mathcal{N} = (G, u(a), \tau(a), k(v), s, d, T)$, $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $d = v_1 \succ \cdots \succ v_k$.
2. Solve LexMCC problem on $\mathcal{N}$ with additional arcs (v_i, s) with capacity $k(v_i)$ and transit times $-(T+1)$ using Algorithm 4.4.3.
3. Construct extended set $\Gamma^E_{v_i}$ as in Algorithm 5.3.1.
4. Push as much flow as possible along each path in $\Gamma^E_{v_i}$ as long as possible within T.
5. Obtain dynamic $s - v_i$ flow on $\mathcal{N}$.

$v_i \in \mathcal{S}$ in given priority order. Therefore, a maximum flow at each $v_i \in \mathcal{S}$ is obtained at the termination of algorithm, [43]. Moreover, the network $\mathcal{N}$ being a two terminal series parallel in structure, this flow has an earliest arrival property due to Theorem 4.4.4.

The extended set of paths $\Gamma^E_{v_i}$ in step 3 of algorithm is constructed by applying the MCC Algorithm 4.4.3 with time complexity of order $O(mn + m\,logm)$ at most nm times for each vertex $v_i \in \mathcal{S}$. Step 4 is executed in constant time for each of at most n vertices $v_i \in V$. Therefore, Algorithm 5.4.1 yields a lexicographic earliest arrival flow on $\mathcal{N}$ in strongly polynomial time. ∎

5.5 Lexicographic Quickest Flow Problem on UPL Network

Consider the network $\mathcal{N}$ with fixed vertex holding capacities. Let us impose the condition for these capacities to be fulfilled as an upper bound as well as a lower bound by the total flow value that is supposed to be held at that vertex. Then vertex capacities can be taken as demands, say, $\mu(v)$ at v for $v \in V \backslash \{s\}$. This consideration allows to see a dynamic flow problem on $\mathcal{N}$ with demands at vertices and asking for a minimum time to satisfy these demands in given priority order. In the following, we define this problem formally. The application of the problem on evacuation planning is obvious when evacuees at the source are known in advance and one wishes to send them to different prioritized safety places of fixed holding capacities.

Given a network $\mathcal{N} = (G, u(a), \tau(a), \mu(v))$ with terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \dots, v_k\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_k$ such that $\sum_i \mu(v_i) = -\mu(s)$ where $\mu : V \to \mathbb{N}_0$ is the demand at the vertex $v \in V$. The negative demand at the source s is termed as supply. Moreover, we restrict the arc capacity $u(a)$ for each arc $a \in A$ to be strictly positive and consider the network $\mathcal{N}$ in such a way that the source vertex s is the *mother vertex* for all the vertices $v_i \in \mathcal{S}$. Then the *lexicographic quickest flow (LexQF) problem* finds a feasible dynamic flow f_{v_i} of given value $\mu(v_i)$ on network $\mathcal{N}$ from the source s to the vertex v_i, in given priority order, which sends the given $\mu(v_i)$ units of flow from s to v_i in minimum number $T(\mu(v_i))$ of time units. Moreover, the excess $ex_f(v_i, T(\mu(v_i)))$ at each $v_i \in \mathcal{S}$ given by equation 5.1 should be equal to the demand at $\mu(v_i)$. That is,

$$ex_f(v_i, T(\mu(v_i))) = \mu(v_i) \;\; \forall \;\; v_i \in \mathcal{S}. \tag{5.4}$$

Thus, the objective function of the lexicographic quickest flow evacuation planning problem asks to lexicographically minimize the vector $(T(\mu(v_1)), T(\mu(v_2)), \dots, T(\mu(v_k))^\top$.

5.5.1 Existence of lexicographic quickest flow

The existence of lexicographic quickest flow on a UPL network $\mathcal{N}$ is obvious, if $\Gamma^E_{v_i}$ (cf. Subsection 5.3.1) is not empty for all $v_i \in \mathcal{S}$. There always exists at least one path from the source s to the vertex v_i in the extended set of paths $\Gamma^E_{v_i}$ with corresponding positive flow value since every vertex $v_i \in \mathcal{S}$ being reachable from s and $u(a)$ being positive for each $a \in A$. This is ensured from the fact that during the construction of extended set $\Gamma^E_{v_i}$ the renovation of the network $\mathcal{N}_{\Gamma_{v_k}}$ with respect to at least one path $\gamma_{v_{i-1}} \in \Gamma^E_{v_{i-1}}$ makes the renovated network free to exist at least one path from s to v_i. Thus, $\Gamma^E_{v_i} \neq \phi$ for each $v_i \in \mathcal{S}$.

5.5.2 Solution for lexicographic quickest flow problem

Here we discuss the solution procedure for the lexicographic quickest flow problem for a UPL network $\mathcal{N}$. The procedure to the problem is similar to the binary search method for solving a quickest flow problem in [24]. In this method, for an strictly increasing sequence of integral time points $\{T_n\}$, an initial interval $I_0 = [T_l, T_u]$ such that $f(T_l) < \mu(v_i) < f(T_u)$, is taken. Here, $f(T_l)$ and $f(T_u)$ denote the maximum dynamic flow value for time horizon T_l and T_u, respectively. Clearly, $T_l \leq T(\mu(v_i)) \leq T_u$ where $T(\mu(v_i))$ is the minimum time that requires for flow units of value $\mu(v_i)$ to send from the source to the vertex v_i. Then the mid-point, say, T_m, of the interval I_0 is computed and $f(T_m)$ is checked for whether it is equal to, less than or greater than $\mu(v_i)$. Depending upon this value, it is decided whether the procedure ends, or should work on the next interval on the left or right of the mid-point T_m.

Since we are interested in finding such minimum time T_m for each vertices $v_i \in \mathcal{S}$ in a priority order, the maximum flow computation technique developed in Section 5.3 is adopted as a subroutine of the procedure with necessary modification. During the procedure, the major step is to construct extended set of paths $\Gamma^E_{v_i}$. Here, the free time intervals I_1 and I_2, if exist, with respect to each path $\gamma_{v_{i-1}} \in \Gamma_{v_{i-1}}$ are to be calculated in each of new selections of mid-point time T_m before renovation of the network $\mathcal{N}_{\Gamma_{v_k}}$. Now, the following cases arises: If $F_t(\gamma_{v_{i-1}}) < T_m$, replace T by T_m in I_2. If $F_t(\gamma_{v_{i-1}}) - N < T_m < F_t(\gamma_{v_{i-1}}) + 1$, discard I_2. If $I_t(\gamma_{v_{i-1}}) < T_m < F_t(\gamma_{v_{i-1}}) + 1 - N$ discard I_2 and replace $F_t(\gamma_{v_{i-1}}) - N$ by T_m in I_1. And, if $I_t(\gamma_{v_{i-1}}) > T_m$, discard both I_1 and I_2. It is to be noted that Γ_{v_j} for all $j > i$ are discarded until we found Γ_{v_i} that sends all flow $\mu(v_i)$ in time horizon T_m such that it is minimum. We proceed with this procedure for each vertices $v_i \in \mathcal{S}$ in given priority order.

Due to the nature of construction of a maximum flow (cf. Algorithm 5.3.1), maximum flow of value $\mu(v_i)$ obtained for time horizon T_m, could also be possible to find in lesser time horizon T'_m for some vertices v_i. That is, it cannot be guaranteed that the time T_m at which dynamic flow of value $\mu(v_i)$ can be sent to v_i is the minimum time to attain this flow value. One should check whether the same flow value is attained for some lesser time. Thus, the following lemma suggested in [24] for $s - d$ quickest flow problem is true in our case also.

Lemma 5.5.1 [17] *For* $f(T_m) = \mu(v_i)$, T_m *is optimal iff* $f(T_m - 1) < \mu(v_i)\ \forall\, v_i \in \mathcal{S}$.

During the procedure, flow computed by the application of LexMDF Algorithm as a subroutine is optimal due to Theorem 5.3.1. Also, this algorithm runs in strongly polynomial time due to Theorem 5.3.2. The next major step in the procedure is to perform a binary search over time horizon repeatedly. This can be done in strongly polynomial time, see Subsection 3.5.2. Thus, from above algorithmic discussion we can assert that LexQF problem on UPL network $\mathcal{N}$ can be solved optimally in strongly polynomial time.

5.6 Contraflow Problems with Capacitated Vertices

We explained the importance of contraflow approach in evacuation planning and discussed various contraflow problems with their solutions in Chapter 4. Here, we introduce a new aspect of evacuation model designed on network with capacitated vertices by imposing contraflow approach on it. The model has arc reversal capability, and is capable of holding evacuees at temporary shelters at intermediate vertices of given priority. Like in the previous sections of this chapter, the flow model we adopt is based on weak-conservation constraints. Based on this aspect, we study MSF problem, MDF problem and EAF problem, and discuss their solution procedures.

5.6.1 Lexicographic maximum static contraflow problem

Consider a static network $\mathcal{N} = (G, l(a), u(a), k(v), s, d)$ where the terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \ldots, v_k\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_k$, to be given and $k(v)$ the vertex capacity at vertex $v \in \mathcal{S}$. Here, the goal is to solve the LexMSF problem with capacitated vertices on $\mathcal{N}$, if direction of arcs on $\mathcal{N}$ can be reversed. We call this problem on $\mathcal{N}$ a *lexicographic maximum static contraflow (LexMSCF) problem.*

For solution procedure of LexMSCF problem, we modify the solution idea (Algorithm 4.3.1) for solving MSCF problem. To ensure the intermediate holding capability in the solution, and to respect the vertex capacities, the maximum flow computation in Step 3 of Algorithm 4.3.1 is replaced by LexMSF flow computation idea given in Algorithm 5.2.1. The modified procedure that solves LexMSCF problem has been given in Algorithm 5.6.1.

Theorem 5.6.1 [18] *Given a static network $\mathcal{N} = (V, A, l(a), u(a), k(v), s, d)$, source s and terminal set $\mathcal{S} = \{v_1, \ldots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$ and $l(a) = 0$ for all $a \in A$. Then, Algorithm 5.6.1 computes a lexicographic maximum static contraflow on $\mathcal{N}$ optimally in strongly polynomial time.*

Proof: The LexMSF Algorithm computes an optimal static flow for each terminals $v_i \in \mathcal{S}$ as sinks on reduced network $\tilde{\mathcal{N}}$ due to Theorem 5.2.1. Again, due to Lemma 4.3.1, these flows are equivalent to the maximum static contraflows on the input network $\mathcal{N}$.

The time complexity of Algorithm 5.6.1 depends on time complexity of the solution procedure on the reduced network $\tilde{\mathcal{N}}$. Also, it is dominated by solving a LexMSF problem on $\tilde{\mathcal{N}}$ since

Algorithm 5.6.1 LexMSCF Algorithm [18]

1. Given a static network $\mathcal{N} = (V, A, l(a), u(a), k(v), s, d)$, $\mathcal{S} = \{v_1, \ldots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$ and integer inputs.

2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, l(\tilde{a}), u(\tilde{a}), k(v), s, d)$ as in Algorithm 4.3.1 and set $l(\tilde{a}) = 0 \ \forall \ \tilde{a} \in \tilde{A}$.

3. Solve LexMSF problem on network $\tilde{\mathcal{N}}$ using Algorithm 5.2.1.

4. Perform flow decomposition into path and cycle flows of maximum flows obtained from step-3 and remove all cycle flows.

5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.

6. Obtain LexMSCF solution on $\mathcal{N}$.

---

the flow decomposition in each iteration and network transformation can be done only in $O(mn)$ ([2]) and $O(m)$ time, respectively. The LexMSF problem can be solved in strongly polynomial time, see Theorem 5.2.1. ■

5.6.2 Lexicographic maximum dynamic contraflow problem for multi-network

We discussed maximum dynamic contraflow problems modeled on multi-network without capacitated vertices in Subsection 4.5.1. Multi-networks capture the situation of road topology with parallel lanes of different transit time and anti-parallel lanes of unequal to-and-fro transit time. Here, we study the problem on multi-network with capacitated vertices for discrete time setting. Since the flow model we adopt here is based on weak-conservation constraints, the model allows holding of flow units at intermediate vertices.

Consider a dynamic multi-network $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(v), s, d, T)$ where the terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \ldots, v_k\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_k$, to be given. We also assume that for every arc $(v, w) \in A$ there exists an arc $(w, v) \in A$. For T being discretized, the goal is to solve the discrete time LexMDF problem given in Section 5.2 on $\mathcal{N}$, if direction of arcs on $\mathcal{N}$ can be reversed. We call this problem on multi-nework $\mathcal{N}$ a *discrete time lexicographic maximum dynamic contraflow (DT-LexMDCF) problem.*

The solution technique for solving DT-LexMDCF problem when arc reversibility on $\mathcal{N}$ is allowed only once at time zero has two major steps: network modification and LexMDF computation. That is, we modify DT-MDCF algorithm for multi-network (cf. Algorithm 4.5.1). The modification is the replacement of maximum flow computation in Step 4 by LexMDF algorithm (cf. Algorithm 5.2.2). The modified procedure is presented in Algorithm 5.6.2.

Algorithm 5.6.2 DT-LexMDCF Algorithm for Multi-network [18]

---

1. Given a dynamic multi-network $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(a), s, d, T)$, $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$ and integer inputs.

2. Transform $\mathcal{N}$ into undirected multi-network $\tilde{\mathcal{N}} = (V, \tilde{A}, l(\tilde{a}), u(\tilde{a}), \tau(\tilde{a}), k(v), s, d, T)$ as in Algorithm 4.5.1 and set $l(\tilde{a}) = 0 \ \forall \ \tilde{a} \in \tilde{A}$.

3. Label each parallel arcs $(v, w) \in \tilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$.

4. Compute LexMDF on network $\tilde{\mathcal{N}}$ using Algorithm 5.2.2.

5. Perform flow decomposition into path and cycle flows of maximum flows obtained from step-4 and remove all cycle flows.

6. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.

7. Obtain DT-LexMDCF solution for multi-network $\mathcal{N}$.

Corollary 5.6.1 [18] *Given a multi-network $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(v), s, d, T)$, source s and terminal set $\mathcal{S} = \{v_1, \ldots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$. Then, Algorithm 5.6.2 computes a discrete time lexicographic maximum dynamic contraflow on $\mathcal{N}$ in pseudo-polynomial time.*

Proof: Let $\mathcal{N}$ be a dynamic multi-network. A maximum dynamic flow computed during the application of Algorithm 5.6.2 on reduced network $\tilde{\mathcal{N}}$ is equivalent to a maximum dynamic contraflow on $\mathcal{N}$ for each iteration, see Theorem 4.5.1. Also, due to Corollary 5.2.1, the DT-LexMDF problem can be solved optimally on $\tilde{\mathcal{N}}$ only in pseudo-polynomial time. ■

5.6.3 Lexicographic earliest arrival contraflow problem for UPL-TTSP network

We discussed about earliest arrival contraflow problems with solution procedures to them modeled on ordinary network as well as a multi-network in Chapter 4. However, these problems were modeled over networks without capacitated vertices. Here, we study the problem modeled over an ordinary network with capacitated vertices based on weak-conservation constraints flow model. Unlike in the case of maximum flow problems, we consider sufficient vertex capacities for some special intermediate vertices.

Consider a dynamic UPL-TTSP network $\mathcal{N} = (V, A, u(a), \tau(a), k(v), s, d, T)$ where the terminal set $\mathcal{S} \subset V$ with $\mathcal{S} := \{v_1, \ldots, v_k\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_k$, to be given. We restrict the network $\mathcal{N}$ with only equal transit times on anti-parallel arcs on it. Moreover, we assume that the vertex v_1 always gets sufficient storage capacity whereas vertices v_i for $i \neq 1$, get either zero or sufficient storage capacities. That is, not all vertices in V have storage capacities. With these settings, for T being discretized, *discrete time lexicographic earliest arrival contraflow (DT-LexEACF) problem* on $\mathcal{N}$ aims to maximize the flow units sent to the terminals on $\mathcal{S}$ in given priority order at each time step $t \in \mathcal{T}$, if direction of arcs can be reversed.

We modify the DT-LexEAF algorithm (cf. Algorithm 5.4.1) to propose the solution for DT-LexEACF problem by applying the network transformation strategy in Algorithm 4.4.1. The modified procedure is given in Algorithm 5.6.3 that solves DT-LexEACF problem when arc reversibility on $\mathcal{N}$ is allowed only once at time zero.

We make the following claim which turns out to be important in proving the optimality of solution computed by Algorithm 5.6.3.

Claim 5.6.1 [66] *UPL-TTSP network $\mathcal{N}$, after transforming into its auxiliary network $\tilde{\mathcal{N}}$, remains UPL-TTSP network.*

Theorem 5.6.2 [66] *Given a UPL-TTSP network $\mathcal{N} = (V, A, u(a), \tau(a), k(v), s, d, T)$, source s and terminal set $\mathcal{S} = \{v_1, \ldots, v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$. Then, Algorithm 5.6.3 computes a discrete time lexicographic earliest arrival contraflow on $\mathcal{N}$ in polynomial time.*

Proof: The statement follows due to Theorems 4.5.4 and 5.4.1, and Claim 5.6.1. ■

Algorithm 5.6.3 DT-LexEACF Algorithm [66]

1. Given a UPL-TTSP network $\mathcal{N} = (G, u(a), \tau(a), k(v), s, d, T)$, $\mathcal{S} = \{v_1, \ldots, v_k\}$ with $d = v_1 \succ \cdots \succ v_k$ and integer inputs.
2. Transform $\mathcal{N}$ into network $\tilde{\mathcal{N}} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), k(v), s, d, T)$ as in Algorithm 4.4.1.
3. Compute LexMDF on $\tilde{\mathcal{N}}$, using Algorithm 5.4.1.
4. Perform flow decomposition into path and cycle flows of maximum flow obtained from step-3 and remove all cycle flows.
5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
6. Obtain DT-LexEACF solution for UPL-TTSP network $\mathcal{N}$.

5.7 Extension of Solutions to Problems in Continuous Time Setting

The continuous time dynamic flow model with weak-conservation constraint can also be defined in similar way as given in Subsection 5.2.1 by integrating the flow units at each time point $t \in [0, T)$ instead of adding flow over time for each time step $t \in \mathcal{T}$ where time horizon is discretized as in the flow conservation case. We discussed in Chapters 3 and 4 that the discrete dynamic flows can be transformed into continuous dynamic flows using the notion of natural transformation. Applying this notion, the solution procedures designed for problems with discrete time setting can also be used to solve them with continuous time setting. Thus, the LexMDF problem with continuous time setting can be solved for general network with integer inputs by applying time expanded network and for UPL network by applying temporally repeated flows. The former problem can be solved only in pseudo polynomial time whereas the latter problem in strongly polynomial time. LexEAF problem with continuous time setting for UPL-TTSP network can similarly be defined, and solved in strongly polynomial time.

Due to the same arguments as above, the contraflow problems with capacitated vertices; namely, DT-LexMDCF problem and DT-LexEACF problem can be extended into one with continuous time setting. We saw in Chapter 4 that the dynamic contraflow problems on ordinary networks with equal transit times on anti-parallel arcs are the particular cases of the dynamic contraflow problems on multi-network without necessarily equal transit times. Therefore, the former problems with capacitated vertices can also be solved by the methods discussed in this section above.

CHAPTER 6

SUMMARY AND CONCLUSIONS

Network flow optimization problems are applicable in modeling real world evacuation scenarios. In this thesis, we discussed some basic network flow problems and extended them to model the evacuation planning problems. Study of maximum dynamic contraflow problem with continuous time setting, modification of network contraflow problems to the problems over multi-network, consideration of capacitated vertices of given priority order in the model to allow holding of flow units at them and application of contraflow approach in the evacuation models with capacitated vertices are our extensions. We discussed solution algorithms for the various evacuation planning problems modeled with these variants.

Disasters caused by natural as well as man-made hazards can have a distressing impact on socio-economic sector including the huge loss of lives and injury of people, and that is why it is crucial to have a disaster management strategy in place. PPRR plan; referring to prevention, preparedness, response and recovery, is widely adopted to lessen the impact of disasters. An efficient evacuation plan plays a crucial role during PPRR plan, particularly, during preparedness and response phases. An urgent escape of people away from an area with threat of disaster to save lives is an evacuation. Besides the introduction of PPRR plan, we highlighted the importance of evacuation planning problem as a primary response immediately after disaster. It is suggested to develop near to real world evacuation plan in advance, and carry out the rehearsal of it in community level for better performance at the time of hazards. Implementation of optimization models is often challenging in real evacuations due to their inexact nature. Unplanned road networks, uncertainty of the population density, the volumes of traffic, operating cost, etc. greatly affect the overall evacuation.

Importance and applicability of the idea of contraflow, especially in evacuation planning problem, has been increasing. Due to its lane-direction reversal property, the idea can be taken as an immediate potential remedy to mitigate congestion by increasing the outbound evacuation route capacity. Besides during emergencies, the approach is equally important to mitigate day to day traffic congestion in crowded urban areas, and to manage one-way mass movements. We discussed network flow optimization models based on this approach found in literature with their solution strategies. In particular, we discussed the maximum dynamic contraflow (MDCF) problem [89] and extended it into one with continuous time setting, a more realistic flow model. Also, we gave its solution technique that runs in strongly polynomial time when arcs are flipped only once at time zero. The notion of natural relation between discrete dynamic flow and continuous dynamic flow plays major role in its solution idea.

We also extended the evacuation model based on contraflow approach into one with more general setting on network. Existing model fails to capture the situation of road topology with parallel lanes of different transit time and anti-parallel lanes of different to-and-fro transit time. The networks we considered for extension are one with not necessarily equal transit time on anti-parallel arcs and another a multi-network. However, the transit time parameter behaves symmetrically as in the existing model. We proposed network modification strategy to give polynomial time algorithms as solutions for evacuation planning problems; namely, MDCF problem and EACF problem with discrete as well as continuous time setting. The MDCF problems are considered on general networks whereas the EACF problems are considered on TTSP networks. EACF problems are unsolved so far with exact solution in polynomial time for general networks as the ordinary EAF problems are in the same situation. In each of the problems arc reversibility has been allowed only once at time zero.

During evacuation, traffic capacity may be dynamically changed due to sudden interruptions or unavailability of roads as the disaster changes (e.g., earthquake and its consequences: damages of bridges or road sections, flooding, landslides, etc.). Thus, consideration of these changes in the model and searching of their solution with optimal selection of dynamic lane reversal would be future works. Searching of solutions to the problems at which arc reversibility is permitted at any time point within the specified time horizon would be immediate research topic. Such model would also be useful in managing every day to and fro traffic flow in crowded cities. We expect that the task could be more appropriate, however, challenging, in continuous time setting due to uncertainty of the interval of time over given time horizon when arcs are to be reversed. Moreover, investigation of the problem with partial contraflow and source to sink path flipping instead of flipping only arcs widens the domain with this approach.

Another important variant of evacuation model we observed is the consideration of holding evacuees at intermediate shelters in it. Evacuation plans with existing model allow evacuees to leave the risk zone only if they can reach the specified destination. Those evacuees which cannot be sent into the destination due to lane capacity limitation can be kept in temporary shelters at intermediate safer places in many real world evacuation scenarios. Thus, evacuators must attempt to send out as many evacuees as possible to safer places, despite the sink, where the evacuees can be provided medical aids or other necessary supports or set priorities among the evacuees to send to the sink.

We presented the model of evacuation planning problem at which flow conservation constraint for vertex has been relaxed. In contrast to existing models, we assumed that the safety terminals (sink and intermediate vertices) are prioritized and have a capacity delimiting the amount of flow that can be held at these spots within given time horizon. With respect to these assumption we solved the lexicographic maximum flow problem on general network with capacitated vertices. The solution algorithm runs in strongly polynomial time for the problem in static version, whereas it takes pseudo-polynomial time for dynamic version. We investigated a strongly polynomial time solution based on temporally repeated flow for dynamic case of the problem modeled on restricted class of network known as uniform path length network. We extended these results to solve lexicographic earliest arrival flow problem modeled on uniform path length two terminal series parallel network with sufficient vertex capacities. Moreover, we considered lexicographic quickest flow problem and discussed its solution that runs in strongly polynomial time.

Moreover, we extended evacuation flow model by incorporating contraflow approach in the model with capacitated vertices. We discussed solution techniques for some of contraflow evacuation problems that we studied in Chapter 4. In particular, we proposed a polynomial time solution for static version of maximum contraflow problem, whereas a pseudo-polynomial one for dynamic version of the problem modeled on multi-network. We considered ordinary network with capacitated vertices for earliest arrival contraflow problem and proposed a polynomial time solution to it.

Investigation of efficient solution to the problems with more general network setting would extend the domain of evacuation planning problem with capacitated vertices. In particular, network flow models with flow dependent (load dependent and inflow dependent) transit time are likely to have potential to capture day-to-day congested traffic as well as one-way mass movement besides the emergency routing for evacuation. Thus, one would incorporate these variants in evacuation model with capacitated vertices as well as in contraflow models for multi-network. Moreover, one can address the transportation system with variety of vehicles, as most of developing cities are experiencing, by extending these models with multi-commodity flow problems.

REFERENCES

[1] Adhikari IM, Pyakurel U and Dhamala TN (2020). An integrated solution approach for the time minimization evacuation planning problem, *International Journal of Operations Research* **17**(1) 27–39.

[2] Ahuja RK, Magnati TL and Orlin JB (1993) *Network flows: theory, algorithms, and applications*, Prentice Hall, New Jersey.

[3] Alipour H and Mirnia K (2017) Uncertain dynamic network flow problems, *Journal of Uncertainty Analysis and Applications* **5**(1) 1–13.

[4] Anderson EJ, Nash P and Philpott AB (1982) A class of continuous network flow problems, *Mathematics of Operations Research* **7**(4) 501–514.

[5] Anderson EJ and Philpott AB (1994) Optimisation of flows in networks over time, *Probability, Statistics and Optimisation* **27** 369–382.

[6] Aronson JE (1989) A survey of dynamic network flows, *Annals of Operations Research* **20**(1) 1–66.

[7] Baumann N (2007) *Evacuation by earliest arrival flows*, Doctoral thesis, Department of Mathematics, University of Dortmund, Germany.

[8] Baumann N and Skutella M (2006) Solving evacuation problems efficiently: earliest arrival flows with multiple sources, In *2006* 47^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* 399–410, IEEE.

[9] Baumann N and Skutella M (2009) Earliest arrival flows with multiple sources, *Mathematics of Operations Research* **34**(2) 499–512.

[10] Bein WW, Brucker P and Tamir A (1985) Minimum cost flow algorithms for series-parallel networks, *Discrete Applied Mathematics* **10**(2) 117–124.

[11] Bhandari PP and Khadka SR (2019) Maximum flow evacuation planning problem with non-conservation flow constraints at the intermediate nodes, In *International Conference on Mathematical Optimization* (8-13 April 2019, Beijing). Retrieved on 12 April 2020 from: `https://www.researchgate.net/publication/332344781`

[12] Bhandari PP and Khadka SR (2019) Contraflow evacuation planning problems for disaster management, In 2^{nd} *International Conference on Earthquake Engineering and Post Disaster Reconstruction Planning*, Bhaktapur, 31–36.

[13] Bhandari PP and Khadka SR (2020a) Evacuation planning problems with intermediate storage, In *International Conference on Applied Mathematics & Computational Sciences (ICAMCS-2019), Dehradun*, 90–95.

[14] Bhandari PP and Khadka SR (2020b) Evacuation contraflow problems with not necessarily equal transit time on anti-parallel arcs, *American Journal of Applied Mathematics* **8**(4) 230–235.

[15] Bhandari PP and Khadka SR (2020c) Maximum flow evacuation planning problem with non-conservation flow constraint, *International Annals of Science* **10**(1) 25–32.

[16] Bhandari PP and Khadka SR (2020d) Maximum contraflow evacuation planning problems on multinetwork, (Submitted)

[17] Bhandari PP and Khadka SR (2020e) Evacuation planning problems on uniform path length network with prioritized destinations, (Submitted)

[18] Bhandari PP and Khadka SR (2021) Lexicographically maximum contraflow problems with vertex capacities, *International Journal of Mathematics and Mathematical Sciences*, **2021**.

[19] Bhandari PP, Khadka SR, Ruzika S and Schäfer LE (2020) Lexicographically maximum dynamic flow with vertex capacities, *Journal of Mathematics and Statistics* **16**(1) 142–147.

[20] Bish DR (2011) Planning for a bus-based evacuation, *OR spectrum* **33**(3) 629–654.

[21] Borradaile G, Klein PN, Mozes S, Nussbaum Y and Wulff-Nilsen C (2017) Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time, *SIAM Journal on Computing* **46**(4) 1280–1303.

[22] Borrmann A, Kneidl A, Köster G, Ruzika S, and Thiemann M (2012) Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models, *Safety science* **50**(8) 1695–1703.

[23] Brucker P (1982) Network flows in trees and knapsack problems with nested constraints, In 8^{th} *Conference on Graph Theoretic Concepts in Computer Science* 25–35.

[24] Burkard RE, Dlaska K and Klinz B (1993) The quickest flow problem, *ZOR - Methods and Models of Operations Research* **37**(1) 31–58.

[25] Bunnagel U, Korte B and Vygen J (1998) Efficient implementation of the Goldberg-Tarjan minimum-cost flow algorithm, *Optimization Methods and Software* **10**(2) 157–174.

[26] Busacker RG and Satty TL (1965) *Finite graphs and networks: an introduction with applications*, McGraw-Hill Book Company.

[27] Chen YL and Chin YH (1990) The quickest path problem, *Computers and Operations Research* **17**(2) 153–161.

[28] Cheriyan J and Maheshwari SN (1989) Analysis of preflow push algorithm for maximum network flow, *SIAM Journal on Computing* **18**(6) 1057–1086.

[29] Cherkassky BV and Goldberg AV (1997) On implementing the push–relabel method for the maximum flow problem, *Algorithmica* **19**(4) 390–410.

[30] Dhamala TN (2015) A survey on models and algorithms for discrete evacuation planning network problems, *Journal of Industrial & Management Optimization* **11**(1) 265–289.

[31] Dhamala TN and Pyakurel U (2013) Earliest arrival contraflow problem on series-parallel graphs, *International Journal of Operations Research* **10**(1) 1–13.

[32] Dhungana RC and Dhamala TN (2019) Maximum FlowLoc problems with network reconfiguration, *International Journal of Operations Research* **16**(1) 13–26.

[33] Dhungana RC, Pyakurel U and Dhamala TN (2018) Abstract contraflow models and solution procedures for evacuation planning, *Journal of Mathematics Research* **10**(4) 89–100.

[34] Dinic EA (1970) Algorithm for solution of a problem of maximum flow in networks with power estimation, *Soviet Mathematics Doklady* **11** 1277–1280.

[35] Dulebenets MA, Pasha J, Abioye OF, Kavoosi M, Ozguven EE, Moses R, Boot WR and Sando T (2019) Exact and heuristic solution algorithms for efficient emergency evacuation in areas with vulnerable populations, *International Journal of Disaster Risk Reduction* **39** 101114.

[36] Edmonds J and Karp RM (1972) Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the ACM (JACM)* **19**(2) 248–264.

[37] Farahani R and Rezapour S (2011) *Logistics operations and management: concepts and models*, Elsevier.

[38] Fleischer L (2001) Faster algorithms for the quickest transshipment problem, *SIAM Journal on Optimization* **12**(1) 18–35.

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

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

[41] Ford LR and Fulkerson DR (1956) Maximal flow through a network, *Canadian Journal of Mathematics* **8** 399–404.

[42] Ford LR and Fulkerson DR (1958) Constructing maximal dynamic flows from static flows, *Operations research* **6**(3) 419–33.

[43] Ford LR and Fulkerson DR (1962) *Flows in networks*, Princeton University Press.

[44] Gale D (1959) Transient flows in networks, *Michigan Mathematical Journal* **6**(1) 59–63.

[45] Gallo G, Grigoriadis MD and Tarjan RE (1989) A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing* **18**(1) 30–55.

[46] Garrett A, Carnahan B, Muhdi R, Davis J, Dozier G, SanSoucie MP, Hull PV and Tinker ML (2006) Evacuation planning via evolutionary computation, In *2006 IEEE International Conference on Evolutionary Computation* 157–164, IEEE.

[47] Goldberg AV and Tarjan RE (1988) A new approach to the maximum-flow problem, *Journal of the ACM (JACM)* **35**(4) 921–940.

[48] Goldberg AV and Tarjan RE (1989) Finding minimum-cost circulations by canceling negative cycles, *Journal of the ACM (JACM)* **36**(4) 873–886.

[49] Goldberg AV and Tarjan RE (1990) Finding minimum-cost circulations by successive approximation, *Mathematics of Operations Research* **15**(3) 430–466.

[50] Göttlich S, Kühn S, Ohst JP, Ruzika S, and Thiemann M (2011) Evacuation dynamics influenced by spreading hazardous material, *Networks & Heterogeneous Media* **6**(3) 443–464.

[51] Grilli ST, Ioualalen M, Asavanant J, Shi F, Kirby JT and Watts P (2007) Source constraints and model simulation of the December 26, 2004, Indian Ocean Tsunami, *Journal of Waterway, Port, Coastal, and Ocean Engineering* **133**(6) 414–428.

[52] Hamacher H, Heller S, Klein W, Köster G and Ruzika S (2011) A sandwich approach for evacuation time bounds, *Pedestrian and Evacuation Dynamics* 503–513, Springer.

[53] Hamacher H and Tjandra S (2001) Mathematical modelling of evacuation problems: A state of art, *Technical Report 24, Fraunhofer (ITWM).*

[54] Hamza-Lup GL, Hua KA, Lee M and Peng R (2004) Enhancing intelligent transportation systems to improve and support homeland security, In 7^{th} *International IEEE Conference on Intelligent Transportation Systems* 250–255.

[55] Hashemi SM and Nasrabadi E (2012) On solving continuous-time dynamic network flows, *Journal of Global Optimization* **53**(3) 497–524.

[56] Hoppe B (1995) *Efficient dynamic network flow algorithms*, Doctoral thesis, Cornell University.

[57] Hoppe B and Tardos E (1994) Polynomial time algorithms for some evacuation problems, In 5^{th} *annual ACM-SIAM symposium on Discrete algorithms* 433–441.

[58] Hoppe B and Tardos E (2000) The quickest transshipment problem, *Mathematics of Operations Research* **25**(1) 36–62.

[59] Iri, M (1996) Network flow: theory and applications with practical impact, In *System Modeling and Optimization* 24–36, Springer, Boston, MA.

[60] Jabari SE, He X and Liu HX (2012) Heuristic solution techniques for no-notice emergency evacuation traffic management, In *Network reliability in practice* 241–259, Springer, New York, NY.

[61] Kamiyama N (2020) Lexicographically optimal earliest arrival flows, *Networks* **75**(1):18–33.

[62] Kelly C (1999) Simplifying disasters: developing a model for complex non-linear events, *Australian Journal of Emergency Management, The* **14**(1), 25–27.

[63] Khadka SR and Bhandari PP (2017a) An efficient solution approach to non-conservation maximum flow evacuation planning problem, In 11^{th} *International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, Malabe, 333–335, IEEE Explore.

[64] Khadka SR and Bhandari PP (2017b) Dynamic network contraflow evacuation planning problem with continuous time approach, *International Journal of Operations Research* **14**(1) 27–34.

[65] Khadka SR and Bhandari PP (2019) Model and solution for non-conservation flow evacuation planning problem, *The Nepali Mathematical Sciences Report* **36**(1-2) 11–16.

[66] Khadka SR and Bhandari PP (2020) Lexicographic earliest arrival contraflow evacuation problem on UPL-TTSP network, *International Journal of Operational Research Nepal* **9**(1) 1–7.

[67] Kim S and Shekhar S (2005) Contraflow network reconfiguration for evacuation planning: a summary of results, In 13^{th} *annual ACM international workshop on Geographic information systems* 250–259.

[68] Kim S, Shekhar S and Min M (2008) Contraflow transportation network reconfiguration for evacuation route planning, *IEEE transactions on Knowledge and Data Engineering* **20**(8) 1115–1129.

[69] Klein, M (1967) A primal method for minimal cost flows with applications to the assignment and transportation problems, *Management Science* **14**(3) 205–220.

[70] Klinz B and Woeginger GJ (2004) Minimum-cost dynamic flows: the series-parallel case, *Networks: An International Journal* **43**(3) 153–162.

[71] Koch R, Nasrabadi E and Skutella M (2011) Continuous and discrete flows over time: a general model based on measure theory, *Mathematical Methods of Operations Research* **73**(3) 301–337.

[72] Kolbe AR, Hutson RA, Shannon H, Trzcinski E, Miles B, Levitz N, Puccio M, James L, Noel JR and Muggah R (2010) Mortality, crime and access to basic needs before and after the Haiti earthquake: a random survey of Port-au-Prince households, *Medicine, Conflict and Survival* **26**, 281–297.

[73] Kotnyek B (2003) An annotated overview of dynamic network flows, *Technical Report, INRIA, Sophia Antipolis* 1–28.

[74] Kovacs P (2015) Minimum-cost flow algorithms: an experimental evaluation, *Optimization Methods and Software* **30**(1) 94–127.

[75] Lin M and Jaillet P (2015) On the quickest flow problem in dynamic networks: a parametric min-cost flow approach, In 26^{th} *annual ACM-SIAM symposium on Discrete algorithms* 1343–1356.

[76] Martins EDQV and Dos Santos JLE (1997) An algorithm for the quickest path problem, *Operations Research Letters* **20**(4) 195–198.

[77] Megiddo N (1974) Optimal flows in networks with multiple sources and sinks, *Mathematical Programming* **7**(1) 97-107.

[78] Megiddo N (1977) A good algorithm for lexicographically optimal flows in multi-terminal networks, *American Mathematical Society* **83**(3).

[79] Megiddo N (1979) Combinatorial optimization with rational objective functions, *Mathematics of Operations Research* **4** 414–424.

[80] Minieka E (1973) Maximal, lexicographic, and dynamic network flows, *Operations Research* **21**(2) 517–527.

[81] Minoux M (2009) On robust maximum flow with polyhedral uncertainty sets, *Optimization Letters* **3**(3) 367–376.

[82] Nojavan M, Salehi E and Omidvar B (2018) Conceptual change of disaster management models: a thematic analysis, *Jamba: Journal of Disaster Risk Studies* **10**(1) 1–11.

[83] Pardalos PM and Arulselvan A (2009) *Multimodal Solutions for Large Scale Evacuations*, Center for Multimodal Solutions for Congestion Mitigation, Department of Industrial and System Engineering, University of Florida.

[84] Philpott AB (1982) *Algorithms for continuous network flow problems*, Doctoral thesis, University of Cambridge, UK.

[85] Philpott AB (1990) Continuous-time flows in networks, *Mathematics of Operations Research* **15**(4) 640–661.

[86] Pyakurel U and Dhamala TN (2015) Models and algorithm on contraflow evacuation planning network problems, *International Journal of Operations Research* **12**(2) 36–46.

[87] Pyakurel U, Dhamala TN and Dempe S (2017) Efficient continuous contraflow algorithms for evacuation planning problems, *Annals of Operations Research* **254** 335–364.

[88] Pyakurel U, Nath HN and Dhamala TN (2019) Partial contraflow with path reversals for evacuation planning, *Annals of Operations Research* **283**(1-2) 591–612.

[89] Rebennack S, Arulselvan A, Elefteriadou L and Pardalos PM (2010) Complexity analysis for maximum flow problems with arc reversals, *Journal of Combinatorial Optimization* **19**(2) 200–216.

[90] Rosen JB, Sun SZ and Xue GL (1991) Algorithms for the quickest path problem and the enumeration of quickest paths, *Computers and Operations Research* **18**(6) 579–584.

[91] Ruzika S, Sperber H and Steiner M (2011) Earliest arrival flows on series-parallel graphs, *Networks* **57**(2) 169–173.

[92] Schmidt M and Skutella M (2014) Earliest arrival flows in networks with multiple sinks, *Discrete Applied Mathematics* **164** 320–327.

[93] Sifaleras A (2013) Minimum cost network flows: problems, algorithms, and software, *Yugoslav Journal of Operations Research* **23** 3–17.

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

[95] Sleator DD and Tarjan RE (1983) A data structure for dynamic trees, *Journal of computer and system sciences* **26**(3) 362–391.

[96] Steiner M (2009) *A survey of earliest arrival flows and a study of the series-parallel case*, Diploma thesis, University of Kaiserslautern.

[97] Telford J and Cosgrave J (2007) The international humanitarian system and the 2004 Indian Ocean earthquake and tsunamis, *Disasters* **31**(1) 1–28.

[98] Tuydes H and Ziliaskopoulos A (2004) Network re-design to optimize evacuation contraflow, In 83^{rd} *Annual Meeting of the Transportation Research Board*, Washington, DC.

[99] Tuydes H and Ziliaskopoulos A (2006) Tabu-based heuristic for optimization of network evacuation contraflow, In 85^{th} *Annual Meeting of the Transportation Research Board*, Washington, DC.

[100] Twigg J (2015) *Disaster risk reduction*, London: Overseas Development Institute, Humanitarian Policy Group.

[101] Urbina E and Wolshon B (2003) National review of hurricane evacuation plans and policies: a comparison and contrast of state practices, *Transportation Research Part A: Policy and Practice* **37**(3) 257-275.

[102] Wei L, Xu J, Lei T, Li M, Liu X and Li H (2019) Simulation and experimental analyses of microscopic traffic characteristics under a contraflow strategy, *Applied Sciences* **9**(13) 2651.

[103] Wilkinson WL (1971) An algorithm for universal maximal dynamic flows in a network, *Operations Research* **19**(7) 1602–1612.

[104] UNDESA (2019) World Urbanization Prospects, the 2018 revision, United Nations, New York.

[105] UNISDR U (2005) Hyogo Framework for Action 2005-2015: Building the resilience of nations and communities to disasters, In *Extract from the final report of the World Conference on Disaster Reduction (A/CONF. 206/6)* (Vol. 380). Geneva: The United Nations International Strategy for Disaster Reduction.

[106] UNISDR U (2009) Terminology on Disaster Risk Reduction, Geneva, Switzerland.

[107] UNISDR U (2015) Sendai framework for disaster risk reduction 2015-2030. In 3^{rd} *UN World Conference on disaster risk reduction*, Sendai, Japan.

[108] Yin D (2009) A scalable heuristic for evacuation planning in large road network, In 2^{nd} *International Workshop on Computational Transportation Science* 19–24.

[109] Zheng YJ, Chen SY and Ling HF (2015) Evolutionary optimization for disaster relief operations: a survey, *Applied Soft Computing* **27** 553–566.

[110] Zhou X, George B, Kim S, Wolff JM, Lu Q and Shekhar S (2010) Evacuation Planning: A Spatial Network Database Approach, *IEEE Data Eng. Bull.* **33**(2) 26–31.

[111] April 2015 Nepal earthquake, In: *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake Assessed on (April 27, 2020).

[112] WHO (May 25, 2020) Coronavirus disease 2019 (COVID-19) situation report – 126. Retrieved from https://www.who.int/docs/default-source/coronaviruse/situation-reports/\20200525-covid-19-sitrep-126.pdf?sfvrsn=887dbd66_2 Assessed on (May 26, 2020).

Appendix

Publications

1. Bhandari PP and Khadka SR (2017). Efficient solution approach to maximum flow evacuation planning problem without flow conservation aspect, *Journal of the Institute of Engineering*, 13(1), 108-116.

2. Bhandari PP and Khadka SR (2018). Non-conserving flow aspect of maximum dynamic flow problem, *Journal of the Institute of Engineering*, 14(1), 107-114.

3. Bhandari PP and Khadka SR (2019) Contraflow evacuation planning problems for disaster management, *Proceedings in* 2^{nd} *Int'l Conference on Earthquake Engineering and Post Disaster Reconstruction Planning*, 25-27 April, 2019, Bhaktapur, 31-36.

4. Bhandari PP and Khadka SR (2020a) Evacuation planning problems with intermediate storage, In *Proceedings of Int'l Conference on Applied Mathematics & Computational Sciences*, 90-95, AIJR Publisher. DOI: https://doi.org/10.21467/proceedings.100.9

5. Bhandari PP and Khadka SR (2020b) Evacuation contraflow problems with not necessarily equal transit time on anti-parallel arcs, *American Journal of Applied Mathematics* **8**(4) 230-235.

6. Bhandari PP and Khadka SR (2020c) Maximum flow evacuation planning problem with non-conservation flow constraint, *International Annals of Science* **10**(1) 25–32.

7. Bhandari PP and Khadka SR (2020d) Maximum contraflow evacuation planning problems on multi-network, (Submitted)

8. Bhandari PP and Khadka SR (2020e) Evacuation planning problems on uniform path length network with prioritized destinations, (Submitted)

9. Bhandari PP and Khadka SR (2021) Lexicographically maximum contraflow problems with vertex capacities, *International Journal of Mathematics and Mathematical Sciences*, **2021**.

10. Bhandari PP, Khadka SR, Schäfer LE and Ruzika S (2020) Lexicographically maximum dynamic flow with vertex capacities, *Journal of Mathematics and Statistics* **16**(1) 142–147.

11. Khadka SR and Bhandari PP (2017a) Dynamic network contraflow evacuation planning problem with continuous time approach, *International Journal of Operations Research* **14**(1) 27-34.

12. Khadka SR and Bhandari PP (2017b) An efficient solution approach to non-conservation maximum flow evacuation planning problem, *Proceedings in* 11^{th} *Int'l Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, 6-8 December, 2017, Malabe, 333-335, IEEE Explore.

13. Khadka SR and Bhandari PP (2019) Model and solution for non-conservation flow evacuation planning problem, *The Nepali Mathematical Sciences Report* **36** 11-16.

14. Khadka SR and Bhandari PP (2020) Lexicographic earliest arrival contraflow evacuation problem on UPL-TTSP network, *International Journal of Operational Research Nepal* **9**(1) 1–7.

Conferences/Workshops/Seminars

1. Non-Conservation Flow Constraint Evacuation Planning Problems. International Conference on Applied Mathematics and Computational Sciences (ICAMCS-2019) (17-19 October 2019, Dehradun).

2. Earliest Arrival Flows on UPL-TTSP Network with Non-Conservation Flow Constraint. 2nd International Conference on Applications of Mathematics to Nonlinear Sciences (AMNS-2019) (27-30 June 2019, Pokhara).

3. Contraflow Evacuation Planning Problems for Disaster Management. International Conference on Earthquake Engineering and Post Disaster Reconstruction Planning (25-27 April 2019, Bhaktapur).

4. Maximum Flow Evacuation Planning Problem with Non-Conservation Flow Constraint at the Intermediate Nodes. International Conference on Mathematical Optimization (11-13 April 2019, Beijing).

5. On Network Contraflow Evacuation Planning Problem. 7th National Conference on Mathematics and Its Applications (NCMA-2019) (12-15 January 2019, Butwal).

6. Evacuation Planning Problem with Non-Conservation Flow Constraints. 9th Nepal Geological Congress (NGC-IX) (19-21 November 2018, Kathmandu).

7. On Maximum Dynamic Flow Problem with Non-Conservation Flow Constraints. 2nd International Conference on Advances in Computational Mathematics (ICACM-2018) (23-24 December 2018, Kathmandu).

8. Evacuation Planning Problem Based on Non-Conserving Flow Model. 11th Triennial Conference of Association of Asia Pacific Operational Research Societies (APORS-2018) (6-9 August 2018, Kathmandu).

9. An Efficient Solution Approach to Non-Conservation Maximum Flow Evacuation Planning Problem. 11th International Conference on Software, Knowledge, Information Management & Applications (06-08 December 2017, Colombo).

10. Non-Conserving Flow Model for the Evacuation Planning Problem. 7th International Science Conference, 10-11 November 2017 at Kathmandu University, Nepal.

11. Maximum Flow Evacuation Planning Problem without Flow Conservation Aspect. National Conference on History and Recent Trends of Mathematics (NCHRTM-2017) (02-04 June 2017, Kathmandu).

12. Non-conserving Flow Model for the Evacuation Planning Problem. AvH workshop on Bilevel Optimization (28 February-07 March 2017, Kathmandu).

13. On Non-conserving Flow Aspect of the Evacuation Planning Problem. National Conference on Mathematics and its Applications (NCMA-2017) (11-13 January 2017, Chitwan).

14. Dynamic Network Contraflow Evacuation Planning Problem with Continuous Time Approach. International conference on Applications of Mathematics to Nonlinear Sciences (AMNS-2016) (26-29 May 2016, Kathmandu).

15. Earliest Arrival Contraflows for Evacuation. International Conference on Earthquake Engineering and Post Disaster Reconstruction Planning (24-26 April 2016, Bhaktapur).

16. Network Optimization for Post-earthquake Disaster Mitigation. Living under Threat of Earthquake International Conference Humboldt Kolleg–Kathmandu (19-22 February 2016).

17. On Maximum Dynamic Network Contraflow Problem in Continuous Time Setting. Seminar: Recent Developments in Mathematics and its Applications- Central Department of Mathematics, Tribhuvan University, Nepal (21-22 September 2015).

International Journal of
Operations Research

International Journal of Operations Research Vol. 14, No. 1, 27−34 (2017)

Dynamic Network Contraflow Evacuation Planning Problem with Continuous Time Approach

Shree Ram Khadka[1] and Phanindra Prasad Bhandari

Central Department of Mathematics, Tribhuvan University,
PO Box 13143, Kathmandu, Nepal

Received March 2016; Revised November 2016; Accepted February 2017

Abstract: A number of efficient algorithms have been established to solve the evacuation problem modeled on dynamic network contraflow approach in discrete-time setting. The arcs are reversed with the consideration of constant transit time and arc capacities over a finite time horizon. In this paper, we consider dynamic network contraflow problem with continuous time setting and propose a strongly polynomial algorithm to solve the maximum dynamic network contraflow evacuation planning problem. Moreover, we propose a pseudo polynomial algorithm for the problem in which the arcs are reversed in any sub-interval of given time horizon.
Keyword — Route Planning, Continuous-time Network Flow, Evacuation Planning.

1. INTRODUCTION

Natural and human-created disasters have been not only causing massive destructions but motivating to a number of researchers to find efficient emergency management procedures also so that the destructions could be minimized. Besides the disasters, it becomes crucial in mass-meetings management and in mitigation of the traffic in a busy traffic hours. An evacuation planning problem asks to find an optimal evacuation plan in a realistic flow model where each evacuee is supposed to be evacuated in a minimal time period. This minimal time period is the lower bound that an evacuee needs.

Evacuation planning is an attempt of sending people and/or their logistics from a dangerous site (source) to a safe site (sink) as quickly as possible. Evacuation planning with lane reversal i.e. contraflow approach designed in discrete time model has been extensively considered in the literature. See, Dhamala (2015), Dhamala and Pyakurel (2013), Kim et al. (2008), Pyakurel and Dhamala (2015) and Rebennack et al. (2010). The contraflow approach reconfigurates the network identifying the ideal direction and reallocating the available capacity for each arc to minimize the evacuation time from source to sink. However, continuous time model naturally better reflects the real world behavior. In continuous time model, flow units can enter the network at any moment of time before the time horizon.

The dynamic network contraflow evacuation planning problem has been formulated as an integer programming formulation by Kim et al. (2008). Two heuristics, one: Greedy heuristic, which determines the condition of congestion and flips highly congested arc in a greedy manner and the other: Bottleneck Relief heuristic, which identifies the bottleneck and increases the capacity by contraflow to improve the maximum flow in each iteration, have also been investigated. The solution is based on empirical results. There exists analytical solution also for the problem which sends a maximum flow from a source to a sink in the two terminal case, see Rebennack et al. (2010). The solution with polynomial complexity has been investigated on both static and dynamic networks where the arc reversal ability has been adapted only once at very beginning of the time horizon for the dynamic case. It is crucial to reverse the arc direction not only at the beginning but at any interval of the time horizon also if the situation of sudden arc damage occurs so that the evacuees must be rerouted for evacuation. The task is more challenging in continuous time setting due to uncertainty of the interval of time over the time horizon when the arc is to be reversed.

The dynamic network flow problem in continuous time setting has been introduced in Philpott (1982). The amount of flow which enters the arc per time unit has been considered to be the flow function. The concept of cuts of source-sink over the time on the network with zero transit time as a solution procedure has been adopted in Anderson et al. (1982). Moreover, the procedure with arbitrary transit time has been extended in Philpott (1990). The amount of flow that arrives at the head of an arc during the unit interval of time beginning at any time step in continuous time setting is equal to the amount of flow that arrives at the head of that arc at that time step in discrete case. This idea has been adapted in Fleischer and Tardos (1998) to transform a feasible flow on a discrete approach

* Corresponding author's e-mail: shreeramkhadka@gmail.com

into a feasible flow on the continuous time setting. Dynamic flow models in continuous time setting can also be found in Koch et al. (2011) and Hashemi and Nasrabadi (2012).

In this paper, we propose an efficient solution approach which produces optimal flow on the dynamic contraflow network in the continuous time setting.

The paper is organized as follows. In Section 2, the mathematical model of the problem is formulated. Section 3 contains the main contribution where a strongly polynomial algorithm on the dynamic contraflow network in which the arc is reversed at time zero is proposed. Moreover, a pseudo polynomial time algorithm for the problem in which the arc can be reversed in any sub-interval of time within the time horizon is also proposed. Section 4 concludes the paper.

2. PROBLEM FORMULATION

Optimization approach of the evacuation planning problem can be described on a network $\mathrm{N} = \left(\mathrm{V}, \mathrm{E}, \mathrm{c_e}, \tau_\mathrm{e}, \mathrm{T}\right)$ where $V\left(\left|V\right| = n\right)$ is for the set of the nodes, $E\left(\left|E\right| = m\right)$ for the set of the arcs joining any two nodes, $\mathrm{c_e}$ for the flow capacity along $\mathrm{e} \in \mathrm{E}$, τ_e for the transit time i.e. the time a flow unit takes along the arc e and T for the time horizon. In particular, the source node and the sink node are denoted as s and d respectively. We denote $\vec{\mathrm{e}} \in E$ for an arc $\left(\mathrm{i}, \mathrm{j}\right)$ in which the flow unit is sent from the node i to the node j and $\overleftarrow{\mathrm{e}} \in E$ for an arc .$\left(\mathrm{j}, \mathrm{i}\right)$. in which the flow unit is sent from the node j to the node i for all $\mathrm{i}, \mathrm{j} \in \mathrm{V}$. Replacement of $\vec{\mathrm{e}}$ by $\overleftarrow{\mathrm{e}}$ is known as the arc reversal. For symmetric transit times, we write $\tau_{\vec{\mathrm{e}}} = \tau_{\overleftarrow{\mathrm{e}}}$ and the auxiliary dynamic network $\tilde{\mathrm{N}} = \left(\mathrm{V}, \tilde{\mathrm{E}}, \mathrm{c}_{\tilde{\mathrm{e}}}, \tau_{\tilde{\mathrm{e}}}, \mathrm{T}\right)$, where $\mathrm{c}_{\tilde{\mathrm{e}}} = \mathrm{c}_{\vec{\mathrm{e}}} + \mathrm{c}_{\overleftarrow{\mathrm{e}}}$, $\tilde{\mathrm{E}} = \left\{\tilde{\mathrm{e}} = \vec{\mathrm{e}} \text{ or } \overleftarrow{\mathrm{e}}\right\}$ and $\tau_{\tilde{\mathrm{e}}} = \tau_\mathrm{e}$ if $\mathrm{e} \in \mathrm{E}$ and $\tau_{\tilde{\mathrm{e}}} = \tau_{\overleftarrow{\mathrm{e}}}$ otherwise. For detail see Example 1. Our network contains no loop and no holdovers at the node. It is note-worthy that evacuation planning problem has been first described in Ford and Fulkerson (1958, 1962). We consider the network on which the flow per time unit i.e. the flow function $\mathrm{f}\left(\mathrm{e}, \theta\right)$ is defined as $f : E \times \left[0, T\right] \to R^+ \cup \left\{0\right\}$, where $\theta \in \left[0, \mathrm{T}\right)$. The flow function satisfies the flow conservation at node $\mathrm{i}, \mathrm{i} \in \mathrm{V}$ if

$$\sum_{e \in \delta^+\left(i\right)} \int_0^T f\left(e, \theta\right) d\theta = \sum_{e \in \delta^-\left(i\right)} \int_0^T f\left(e, \theta\right) d\theta; \forall \theta \in \left[0, T\right), \tag{1}$$

where $\delta^+\left(\mathrm{i}\right) = \left\{\left(\mathrm{j}, \mathrm{i}\right) \in \mathrm{E}\right\} \forall \mathrm{j} \in \mathrm{V}$ for the set of arcs heading towards node i and $\delta^-\left(\mathrm{i}\right)$ $= \left\{\left(\mathrm{i}, \mathrm{j}\right) \in \mathrm{E}\right\} \forall \mathrm{j} \in \mathrm{V}$ for the set of arcs leaving node i. The flow function satisfies the capacity constraints also i.e. $\mathrm{f}\left(\mathrm{e}, \theta\right) \leq \mathrm{c_e}, \forall \mathrm{e} \in \mathrm{E}$ and $\forall\ \theta \in \left[0, \mathrm{T}\right)$. An $\mathrm{s} - \mathrm{d}$ flow in which the flow function satisfies conservation constraint for any intermediate node $\mathrm{i} \in \mathrm{V} \setminus \left\{\mathrm{s}, \mathrm{d}\right\}$ and capacity constrain for all arcs at every time $\theta \in \left[0, \mathrm{T} - \tau_\mathrm{e}\right)$ is said to be a feasible $s - d$ flow where τ_e is the transit time along the arc $\mathrm{e} \in \mathrm{E}$. The flow which obeys conservation constraint at each node $\mathrm{i} \in \mathrm{V}$ is commonly known as a circulation. Obviously, $\mathrm{f}\left(\mathrm{e}, \theta\right) = 0 \quad \forall \theta \notin \left[0, \mathrm{T} - \tau_\mathrm{e}\right)$ and all flow units leave the network before the completion of time horizon T. Let $\mathbf{f}$ be the net flow value that leaves the source over all time steps or enters the sink over all time steps $\theta \in \left[0, \mathrm{T}\right)$.We can describe the net flow as follows.

$$\begin{aligned} \mathbf{f} &= \sum_{e \in \delta^-\left(s\right)} \int_0^T \mathrm{f}\left(e, \theta\right) d\theta - \sum_{e \in \delta^+\left(s\right)} \int_0^T \mathrm{f}\left(e, \theta\right) d\theta \\ &= \sum_{e \in \delta^+\left(d\right)} \int_0^T \mathrm{f}\left(e, \theta\right) d\theta - \sum_{e \in \delta^-\left(d\right)} \int_0^T \mathrm{f}\left(e, \theta\right) d\theta \end{aligned} \tag{2}$$

The net flow $\mathbf{f}$ is maximized for the maximum dynamic network contraflow problem. Let $\mathbf{f_d}$ and $\mathbf{f_c}$ denote the net flow for the discrete and the continuous time setting, respectively. Let the net flow **f** for the static case be $\mathbf{f_s}$.

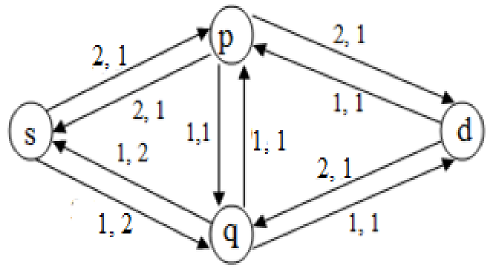


Figure 1. An evacuation network N.

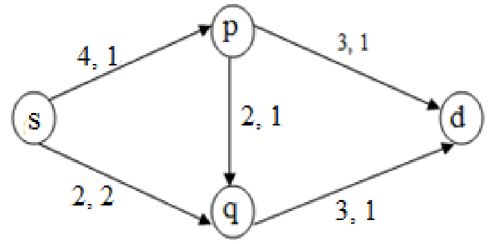


Figure 2. A contraflow reconfigured network $\tilde{N}$ of Fig 1.

Example 1. Consider a two terminal evacuation network as depicted in Fig 1. Line joining any two ovals (the node) is an arc for example a road. Here, arrow on the arc shows the direction of the flow. Node s is the dangerous place (source) that contains evacuees, d is the safe place (sink) that waits them with sufficient capacity and the remaining nodes are the intermediate nodes. The movement of the evacuees (possibly cars) is the flow. The first and the second numbers next to each arc are the arc capacity and the transit times respectively. For example, an arc between nodes s and p (directed towards p) has capacity 2 and transit time 1. That is, 2 vehicles can pass simultaneously through arc (s, p) within 1 unit time. A unit of time may be group of minutes or hours. A contraflow reconfiguration directed towards sink d of the network depicted in Fig 1 is shown in Fig 2. Capacities have been added but the transit time remains unaltered in each arc.

We use the notion of the natural transformation discussed by Fleischer and Tardos (1998) which states that the amount of flow that arrives at the node j through the arc e at time step θ in the discrete approach is equal to the amount of flow arriving at j through the arc e during the unit interval of time at the beginning of time step θ i.e. $\mathbf{f_d}(e,\theta) := \mathbf{f_c}(e,[\theta,\theta+1))$ for all $\theta \in \{0,1,\dots,T-1\}$. Flow movement on the arc with continuous time setting is shown in Fig 3 where the arc $e=(s,q)$ has a capacity $c_e = 2$ (after contraflow reconfiguration) and transit time $\tau_e = 2$ for each unit of flow. Each unit of flow entering the arc at time θ starts reaching at q at time $\theta + \tau_e$, that is, at time $\theta + 2$. The flow totally leaves the arc only at time $\theta + \tau_e + 1$ that is, at time $\theta + 3$. However, it can be considered that the same amount of flow completes its journey at time $\theta + 2$ in discrete time setting.

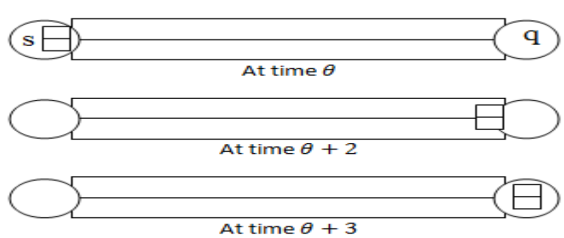


Figure 3. Continuous flow on arc having capacity 2 and transit time 2.

3. MAXIMUM DYNAMIC CONTRAFLOW

There exists a relationship between the maximum flow values of discrete and the continuous net flows while sending the evacuees from the source to the sink. The approach is based on chain decomposition. Let $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_r\}$ be the set of chains with static flow values $v(\gamma_1), v(\gamma_2), \cdots, v(\gamma_r)$ respectively. The static, the discrete and the continuous net flow values are $\mathbf{f_s} = \sum_{k=1}^{r} v(\gamma_k), \mathbf{f_d} = \sum_{k=1}^{r} v(\gamma_k).(T - \tau_{\gamma_k} + 1)$ and $\mathbf{f_c} = \sum_{k=1}^{r} v(\gamma_k).(T - \tau_{\gamma_k})$ respectively. Here, the term τ_{γ_k} stands for the transit time a unit flow takes along the chain γ_k. The natural transformation shows that the net flow value in continuous time setting does not exceed the case in discrete approach. Moreover, the maximum difference of the net flow between continuous and discrete time model within the given time horizon is maximum static flow.

We first consider the maximum dynamic contraflow problem with continuous time setting with arc reversal at time zero then the problem with the arc reversal in any sub-interval of time horizon T.

3.1 Arc Reversal at Time Zero

The problem with arc reversal at time zero in discrete model has been investigated in Rebennack et al. (2010). The problem has been solved with a strongly polynomial time algorithm. Maximum flow is obtained from the chain decomposition starting each chain flow at time zero and then adopting temporal repetition. There exists a temporally repeated flow in the continuous time setting also. Moreover, the flow is maximal due to the following lemma of Anderson and Philpott (1994).

Lemma 1. *[Anderson and Philpott (1994)] The temporally repeated flow with continuous time setting is maximal over the time horizon.*

The investigation of Ford and Fulkerson (1958), which states that a feasible flow on N is an equivalent feasible flow of the problem on the corresponding time expanded network, assures that the problem can be solved by converting the dynamic network N into the time expanded network N^T over the time horizon T. The time expanded network N^T is defined as $N^T = (V^T, E^T, c_e, \tau_e, T)$, where

$$V^T = \{i(\theta) : i \in V \text{ and } \theta \in \{0,1,\ldots,T-1\}\}$$

and

$$E^T = \{(i(\theta), j(\theta + \tau(i,j))) : i \neq j, i, j \in V \text{ and } \theta \in \{0,1,\ldots,T-1-\tau_{(i,j)}\}\}.$$

Lemma 2. *The maximum flow for two terminal case of the maximum dynamic contraflow problem with continuous time setting on N does not exceed the optimal flow for the corresponding time expanded network N^T.*

Proof: Every feasible flow on N is equivalent feasible flow of the maximum dynamic contraflow problem with continuous time setting on the corresponding time expanded network. Furthermore, the continuous time net flow $\mathbf{f_c}$ does not exceed the discrete time net flow $\mathbf{f_d}$.

Now we propose an algorithm say MDNCF-CT which can yield an optimal maximum flow on the dynamic network with arc reversal in continuous time setting. This is a modified algorithm designed for the discrete model in Rebennack et al. (2010).

Algorithm - 1 (Algorithm MDNCF-CT)

1. Transform the network $N = (V, E, c_e, \tau_e, T, \text{integer})$ into $\tilde{N} = (V, \tilde{E}, c_{\tilde{e}}, \tau_{\tilde{e}}, T)$ where $c_{\tilde{e}} = c_{\vec{e}} + c_{\overleftarrow{e}}$, $\tilde{E} = \{\tilde{e} = \vec{e} \text{ or } \overleftarrow{e}\}$ and $\tau_{\tilde{e}} = \tau_e$ if $e \in E$?and $\tau_{\tilde{e}} = \tau_{\overleftarrow{e}}$ otherwise.
2. Compute the discrete dynamic, temporally repeated flow on network $\tilde{N}$ for time horizon $T-1$.
3. Transform the discrete dynamic flow into continuous dynamic flow using the natural transformation $\mathbf{f_d}(e,\theta) = \mathbf{f_c}(e,[\theta,\theta+1))$ for all $\theta \in \{0,1,\cdots,T-1\}$.
4. Perform the flow decomposition into chain and cycle flows of the maximum flow obtained from step-3 and remove all cycle flows.

5. Arc $\overleftarrow{e} \in E$ is reversed if and only if the flow along arc $\vec{e} \in E$ is greater than c_e or if there is non-negative flow along arc $e \notin E$.
6. Obtain the maximum dynamic contraflow with continuous time setting for the given time horizon T.

The algorithm MDNCF-CT yields an optimal solution to the maximum dynamic contraflow problem with continuous time setting. The following is the proof of correctness of the algorithm.

Theorem 1. *The algorithm MDNCF-CT yields an optimal solution to the maximum dynamic contraflow problem with continuous time setting on the network N.*

Proof: The algorithm MDNCF-CT is a modified algorithm P-MDCF investigated in Rebennack et al. (2010). Thus the steps 1, 2 and 4 are clearly well defined. The flow decomposition breaks the optimal flow into chains from source to sink and into cycles with positive flows. These positive flows vanish in each cycles after cancelation and ensures that there is either a flow along arc (i,j) or (j,i), but never on both arcs. Hence the resulting flow from step 5 is a feasible flow with arc reversal for N. Step 3 is feasible since the natural transformation converts of a feasible $(T-1)$-horizon maximum dynamic flow in discrete time setting into a feasible T-horizon maximum dynamic flow in continuous time setting.

Since every feasible flow for the continuous time setting on $\tilde{N}$ is also feasible for the continuous time setting on N, the algorithm is correct for the feasible flow for the continuous time setting.

By the feasibility condition,

$$\tilde{N}_{MDF-CT_{opt}} \leq N_{MDCF-CT_{opt}}$$

where $N_{MDCF-CT_{opt}}$ and $\tilde{N}_{MDF-CT_{opt}}$ stand for the optimal value of the maximum dynamic flow on N and on $\tilde{N}$, respectively in the continuous time setting.

It is clear by Lemma 2 that the maximum static contraflow on N^T is not less than the maximum dynamic contraflow on N in continuous time setting. That is,

$$N_{MDCF-CT_{opt}} \leq N^T{}_{MSCF_{opt},}$$

where $N^T{}_{MSCF_{opt}}$ is the optimal value of the maximum static flow on N^T.

We have the fact that the maximum static contraflow problem on N^T is equivalent to the maximum static flow problem on $\tilde{N}^T$ where the arc set $\tilde{E}$ is defined as $\tilde{E} = \left\{\tilde{e} = \vec{e} \in E \text{ or } \overleftarrow{e} \in E\right\}$, $c_{\tilde{e}}$ is defined as $c_{\tilde{e}} = c_{\vec{e}} + c_{\overleftarrow{e}}$ and the transit time is $\tau_{\tilde{e}} = \tau_e$ if $e \in E$ and $\tau_{\tilde{e}} = \tau_{\overleftarrow{e}}$ otherwise.

Thus,

$$N^T{}_{MSCF_{opt}} = \tilde{N}^T_{MSF_{opt}}$$

where $\tilde{N}^T_{MSF_{opt}}$ stands for the optimal value of the maximum static contraflow on $\tilde{N}^T$.

By Lemma 1, the maximum flow in time expanded network $\tilde{N}^T$ can be obtained by temporally repeated chain flow of a static network $\tilde{N}$. That is, $\tilde{N}^T_{MSF_{opt}} = \tilde{N}_{MDF-CT_{opt}}$.

Hence, we have

$$\begin{aligned} N_{MDCF-CT_{opt}} &\leq N^T{}_{MSCF_{opt}} \\ &= \tilde{N}^T_{MSF_{opt}} \\ &= \tilde{N}_{MDF-CT_{opt}}. \end{aligned}$$

Therefore,

$$\tilde{N}_{MDF-CT_{opt}} = N_{MDCF-CT_{opt}}.$$

The following example illustrates the presented algorithm MDNCF-CT.

Example 2. Consider the evacuation network N of Fig 1 with time horizon $T = 4$. At first we transform the network as described in step 1 and get the transformed network $\tilde{N}$ as shown in Fig 2. With the aid of step 2 we obtain the discrete time maximum dynamic flow of value 9 in $\tilde{N}$ for time horizon $T - 1 = 3$. We calculate the flow in $\tilde{N}$ for each time step $\theta \in \{0,1,2,3\}$. At time steps 0 and 1 no flow reaches at sink. The chain $(s - p - d)$ carries 3 units of flow at sink at time $\theta = 2$ for the first time. The chains $(s - p - d)$, $(s - p - q - d)$ and $(s - q - d)$ respectively carry 3 units, 1 unit and 2 units of flow at sink at time $\theta = 3$. Now, step 3 converts this discrete time flow into continuous time in the following manner: Flow of value 3 at time $\theta = 2$ is considered as the continuous time flow of same amount in the time interval $[2,3)$. Similarly, the flow of value 3, 1 and 2 at time $\theta = 3$ are considered as the continuous time flow of same amounts in the time interval $[3,4)$. Thus summing up these flow values within time horizon $T = 4$ we get the continuous time dynamic flow of value 9. This flow value is the maximum dynamic flow in continuous time setting in $\tilde{N}$ for time horizon T. Eventually, the maximum dynamic contraflow in continuous time setting in N for given time horizon $T = 4$.

Theorem 2. *The Algorithm MDNCF-CT solves the maximum dynamic contraflow problem with continuous time setting in strongly polynomial time.*

Proof: Finding a temporally repeated flow is equivalent to solving a minimum cost flow problem. The algorithm due to Goldberg and Tarjan (1989) leads to a strongly polynomial time of order $O(n^2.m^3.\log n)$ for solving this problem. Let us denote it by $h_1(n,m)$. Since the natural transformation of $(T-1)$ -horizon discrete time maximum dynamic flow yields a T-horizon continuous time maximum dynamic flow, the time complexity of finding a temporally repeated continuous flow is also $h_1(n,m)$. Therefore, time complexity of Algorithm-1 is $O(h_1(n,m) + h_2(n,m)) + h_3(n,m))$ where $h_2(n,m) = O(n^2.\sqrt{m})$ and $h_3(n,m) = O(n.m)$ are the times required to solve the maximum static flow (MSF) problem and the flow decomposition respectively; which is strongly polynomial.

3.2 Arc Reversal at any Sub-interval of the Time Horizon

During evacuation we may encounter the situation with all of sudden blockage of road segments that causes obstacles for evacuees from being evacuated through the current route (chain). The model, allowing the arc reversal capability only once at time zero, cannot deal with such situation. In the following, we have tried to overcome this hurdle, if exists, by rerouting (as we have not considered the immediate road repairing after disaster) the flow unit (evacuee) that is currently traveling on the network by contraflow approach. In this model we do not restrict the arc reversal capability only at time zero but allow reversing, if necessary, in any time interval $[\theta,\theta+1)$ for all $\theta \in \{0,1,\ldots,T-1\}$ only once at the beginning and call it the generalized maximum dynamic contraflow problem with continuous time setting (G-MDNCF-CT). We have proposed solution procedure, Algorithm-2 (Algorithm G-MDNCF-CT) below, to solve this problem.

An arc reversal capability has been considered to be at each integer time points within given time horizon T for lexicographically maximum dynamic contraflow (LMDCF) problem in Pyakurel and Dhamala (2015). The LMDCF is the maximum dynamic contraflow that maximizes the flow in given priority of terminals. An algorithm to solve this problem has been given with polynomial time complexity $O(\delta \times (m \log n (m + n \log n))$ where m, n and δ are the numbers of arcs, nodes and terminals, respectively. However, their model is based on discrete time setting.

We define the term Last-Returns-First property (LRF property) for the situation in which an arc allows to return back for the last flow unit at the first, the second-last flow unit at the second and so on. An arc e is a dead arc at time θ if all or some flow units on it are blocked at time θ. An arc remains dead throughout the time horizon after the time of death if it is dead once as there is no immediate road repairing consideration in our model. Without loss of generality, we assume that the capacity of a dead arc is zero since our model does not allow partial contraflow. In particular, if a chain contains a dead arc at time θ we reverse only those necessary arcs of it which are directed towards the dead arc only once at the beginning of the time interval $[\theta, \theta+1)$. The flow units on every arc of the chain containing a dead arc should satisfy the LRF property while reversing its direction at the time if there are flow units traveling on it. This prevents two flow units (possibly cars) from being collided each other. How we handle such problem is explained in Example 3.

Algorithm - 2 (Algorithm G-MDNCF-CT)

1. Transform the network $N = (V, E, c_e, \tau_e, T, \text{integer})$ into $\tilde{N} = (V, \tilde{E}, c_{\tilde{e}}, \tau_{\tilde{e}}, T)$ where $c_{\tilde{e}} = c_{\vec{e}} + c_{\overleftarrow{e}}$, $\tilde{E} = \{\tilde{e} = \vec{e} \text{ or } \overleftarrow{e}\}$ and $\tau_{\tilde{e}} = \tau_e$ if $e \in E$ and $\tau_{\tilde{e}} = \tau_{\overleftarrow{e}}$ otherwise.
2. Compute the discrete dynamic, temporally repeated flow on network $\tilde{N}$ for time horizon $T-1$.
3. Transform the discrete dynamic flow into continuous dynamic flow using the natural transformation $\mathbf{f_d}(e,\theta) = \mathbf{f_c}(e,[\theta,\theta+1))$ for all $\theta \in \{0,1,\ldots,T-1\}$.
4. Perform the flow decomposition into chain and cycle flows of the maximum flow obtained from step-3 and remove all cycle flows.
5. Arc $\vec{e} \in E$ is reversed if and only if the flow along arc $\vec{e} \in E$ is greater than c_e, if there is non-negative flow along arc $e \notin E$ or arc e is dead at time θ.
6. Obtain the generalized maximum dynamic contraflow with continuous time setting for the given time horizon T.

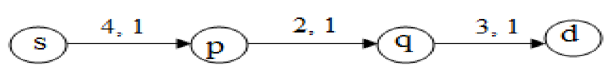


Figure 4. An s-d chain of the evacuation network of Fig 2.

Example 3. Consider an $s-d$ chain of the evacuation network (after contraflow reconfiguration) given in Fig 2 as depicted in Fig 4 with (q,d) a dead arc at time $\theta+2$ among three arcs. Let us start to send any number of flow units less or equal to 2 from s at time θ and continue it until time $\theta+1$. The Flow units that entered the arc (s,p) at time θ reaches at q at time $\theta+2$. But, at this time the arc (q,d) is dead and flow cannot move towards sink d via current route (chain). Now, we reverse the direction of arc (p,q) (i.e. from heading towards q to heading towards p) at the beginning of the time interval $[\theta+2, \theta+3)$ so that all the flow units that has entered on arc (p,q) return back to p by satisfying LRF property. That is, the flow units that entered the arc at the last should return back at the first and so on.

In step 5 of Algorithm G-MDNCF-CT, direction of the arc is allowed to reverse only after satisfying the LRF-property if dead arc do exist and therefore flow in the arc does not travel in both directions at the same time. Other steps are similar to that of Algorithm MDNCF-CT and their feasibility have been discussed already in Theorem-1. Moreover, each direction reversal takes place only at the beginning of the time interval. However, the direction of the same arc may be reversed more than once within the given time horizon T in different time intervals. This can happen at most T times for at most m arcs. Therefore, step 5 of algorithm-2 can be carried out in at most $O(T.m)$ time, depending on T, that dominates the overall time bound of the algorithm. Thus we can state the following theorem:

Theorem 3. *Algorithm G-MDNCF-CT solves generalized maximum dynamic contraflow problem with continuous time setting in pseudo polynomial time.*

4. CONCLUDING REMARK

The importance and applicability of the idea of contraflow especially in evacuation planning problem has been increasing. The model of the problem and the solution approaches based on continuous time setting better reflects the real world situation. In this paper, we have considered the problem with two algorithms as solution procedures with strongly polynomial time complexity if the arcs are flipped only once at time 0 in continuous time setting. Furthermore, an algorithm, with pseudo polynomial time complexity under the consideration of flipping the arcs in any sub-interval of given time horizon, has also been investigated.

Investigation on the problem with partial contraflow and total chain flipping instead of flipping only the arcs would be an interesting research area in the future.

ACKNOWLEDGEMENTS

The research of Shree Ram Khadka was supported by the European Commission in the framework of Erasmus Mundus and within the project cLINK and University of Bremen, Bremen, Germany and that of Phanindra Prasad Bhandari was partially supported by University Grants Commission, Nepal. We are grateful to the two anonymous referees for their thoughtful and insightful comments which significantly improved the paper.

REFERENCES

1. Anderson, E.J., Nash, P. and Philpott, A.B. (1982). A Class of Continuous Network Flow Problems. *Mathematics of Operations Research,* 7: 501-514.
2. Anderson, E.J. and Philpott, A.B. (1994). Optimisation of Flows in Networks over Time. *Probability, Statistics and Optimisation, F. P. Kelly, ed., Wiley, Newyork*, ch. 27, 369-382.
3. Dhamala, T.N. (2015). A Survey on Models and Algorithms for Discrete Evacuation Planning Network Problems. *Journal of Industrial and Management Optimization*, 11: 265-289.
4. Dhamala, T.N. and Pyakurel, U. (2013). Earliest Arrival Contraflow Problem on Series-parallel Graphs. *International Journal of Operations Research*, 10, 1: 1-13.
5. Fleischer, L. and Tardos, E. (1998). Efficient Continuous-time Dynamic Network Flow Algorithms. *Operations Research Letters*, 23: 71-80.
6. Ford, L.R. and Fulkerson, D.R. (1958). Constructing Maximal Dynamic Flows from Static Networks. *Operations research*, 6, 3: 419-433.
7. Ford, L.R. and Fulkerson, D.R. (1962). Flows in Networks. *Princeton University Press.*
8. Goldberg, A.V. and Tarjan, R.E. (1989). Finding Minimum-cost Circulations by Cancelling Negative Cycles. *Journal of ACM*, 36: 873-886.
9. Hashemi, S.M. and Nasrabadi, E. (2012). On Solving Continuous-time Dynamic Network Flows. *J Glob Optim,* 53, 3: 497-524.
10. Kim, S., Shekhar, S. and Min, M. (2008). Contraflow Transportation Network Reconfiguration for Evacuation Route Planning. *IEEE transactions on Knowledge and Data Engineering*, 20, 8: 1115-1129.
11. Koch, R., Nasrabadi, E. and Skutella, M. (2011). Continuous and Discrete Flows over Time: a General Model Based on Measure Theory. *Math Meth Oper Res*, 73: 301-337.
12. Philpott, A.B. (1982). Algorithms for Continuous Network Flow Problems. *PhD thesis, University of Cambridge, UK.*
13. Philpott, A.B. (1990). Continuous-time Flows in Networks. *Mathematics of Operations Research*, 15: 640-661.
14. Pyakurel, U. and Dhamala, T.N. (2015). Models and Algorithm on Contraflow Evacuation Planning Network Problems. *International Journal of Operations Research*, 12, 2: 36-46.
15. Pyakurel, U., Hamacher, H.W. and Dhamala, T.N. (2014). Generalized Maximum Dynamic Contraflow on Lossy Network. *International Journal of Operations Research Nepal*, 3: 27-44.
16. Rebennack, S., Arulselvan, A., Elefteriadou, L. and Pardalos, P.M. (2010). Complexity Analysis of Maximum Flow problem with Arc Reversal. *Journal of Combinatorial Optimization*, 29: 200-216.

The Nepali Math. Sc. Report
Vol.36, No.1 and 2, 2019

MODEL AND SOLUTION FOR NON-CONSERVATION FLOW EVACUATION PLANNING PROBLEM

SHREE RAM KHADKA [1] AND PHANINDRA PRASAD BHANDARI [2]

[1,2] *Central Department of Mathematics, Tribhuvan University,*
PO Box 13143, Kathmandu, Nepal

Abstract: Efficient evacuation plan with which a maximum evacuees can be sent as soon as possible from the disastrous place to the safe place is an important notion during the response phase of the disaster management. Such a plan in terms of optimization models has been extensively studied in a various scenarios, see [3]. The optimization models have been based on the flow conservation constraint which permits an evacuee to be taken out of the disastrous place only if it can be sent into the safe place. However, the evacuation plan model with no flow conservation can keep several evacuees in the relatively safe places besides the evacuees which could be sent into the safe place.

In this paper, we describe an optimization model for the evacuation plan based on the non-conservation flow constraint with an efficient solution procedure which keeps a maximum evacuees on the prioritized intermediate places besides a maximum evacuees into the specified safe place.

Key Words: Network Flow, Flow Conservation, Preflow-Push Algorithm, Evacuation Planning Problem
AMS (MOS) Subject Classification. 90B10, 90B20.

1. Introduction

Disaster management includes prevention which attempts permanent protection from disasters, planning which focuses on preparing the equipment and procedures for the use during the response after the disaster and recovery which attempts to bring the affected area and people back to normalcy. Efficient evacuation planing over the existing road network is an important notion of the planning phase of the disaster management. The main objective of the evacuation planning is to find an efficient procedure so that maximum number of evacuees can be evacuated from the disastrous place, the source, as soon as possible to the safe place, the sink. The procedure can also be useful for the traffic mitigation during the rush hour in a crowd urban area.

Evacuation plan modeled with flow conservation allows evacuees to leave the source only if they can reach the sink. The literature has been flourished with wide range of studies on the problems based on this characteristic since the investigation of two-terminal maximum static flow problem in [4], see the survey articles [9] and [3]. The maximum dynamic flow problem, that maximizes the flow from a source to a sink in given time horizon, has been studied in [5] and [6]. Moreover, minimizing the total time to send the given flow from the source to the sink, known as quickest flow problem and maximizing flow into the sink at

Received Aug. 7, 2019 .

each time step within the time horizon, known as earliest arrival flow problem has also been studied, see [3]. Maximum flow evacuation planning problems with contraflow approach, reversing the direction of the arcs so that flow into the sink can be increased within the specified time horizon, have been studied in [11], [13], [8] and [12].

The evacuation planning model with flow conservation constraint does not allow the evacuees to be kept at the intermediate places. However, the model with the non-conservation constraint allows to keep at the intermediate places besides a maximum number of evacuees to send into the sink. A preliminary solution approach, which is based on the preflow and push algorithm investigated in [7], has been proposed in [1], [2] and [10].

This paper presents a model for the evacuation plan based on no flow conservation concept with an efficient solution procedure that keeps a maximum evacuees on the prioritized intermediate places besides a maximum evacuees into the specified safe place.

2. Problem Formulation

An evacuation planning problem modeled on the network $N = (V, E, c_e, \tau_e, s, d, T)$, see Figure 1, maximizes the total flow of evacuees $\mathbf{f}$ into the specified safe place d, the sink, through the route segments $e \in E$, the arcs, of the routes initiated from the dangerous place s, the source, over the time horizon T. That is,

$$\text{maximize } \mathbf{f} = \sum_{e \in \delta^-(d)} \sum_{\theta=0}^{T} f(e, \theta),$$

where $\delta^-(d)$ denotes the set of arcs entering into the sink d. The flow unit $f(e, \theta) \to R^+ \cup \{0\}$ flows at time step $\theta \in \{0, 1, \ldots, T\}$ along the arc $e = (v, w)$ with τ_e as transit time. Here v and w, the nodes, are the set of intersections of arcs. The flow unit follows the following constraints.

The flow units cannot exceed the arc capacity c_e for any time step, i.e.

$$0 \le f(e, \theta) \le c_e, \quad \text{for all } \ e \in E \text{ and for all } \ \theta \in \{0, 1, \ldots, T\}, \tag{2.1}$$

E being the set of arcs.

The flow units that enters into a node v for each time step may not exit from it at the same time or later within the time horizon T, i.e.

$$\sum_{e \in \delta^-(v)} \sum_{\theta=0}^{T} f(e, \theta) - \sum_{e \in \delta^+(v)} \sum_{\theta=0}^{T} f(e, \theta) \ge 0, \quad \text{for all } \ v \in V - \{s, d\} \tag{2.2}$$

where $\delta^-(v)$ and $\delta^+(v)$ denote for the set of arcs entering into the node v and leaving from it, respectively. The flow units remained at the node v at time $\theta \in \{0, 1, \ldots, T\}$, say the excess flow units $e_v(\theta)$, satisfies

$$e_v(\theta) \le \sum_{e \in \delta^-(v)} f(e, \theta) \ \text{ for all } \ \theta \in \{0, 1, \ldots, T\}. \tag{2.3}$$

Such a node $v \in V - \{s, d\}$ at which $e_v(\theta) > 0$ is said to be an active node at time θ and the corresponding flow $f(e, \theta)$ is a pre-flow.

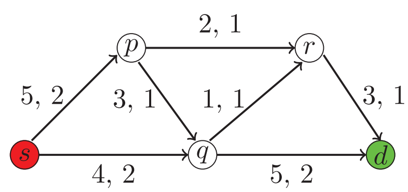


FIGURE 1. An evacuation network N with s is the dangerous place (source) and d is the safe place (sink). The first and the second numbers next to each arc are the capacity and the transit times respectively.

Additionally, it is allowed to hold flow $h_{v'}$ at the temporary shelter v' of vertex $v \in V$ which is given by

$$0 \leq h_{v'} = \sum_{\theta=0}^{T} \sum_{e \in \delta^-(v)} f(e, \theta - \tau_e) - \sum_{\theta=0}^{T} \sum_{e \in \delta^+(v)} f(e, \theta) \tag{2.4}$$

for all $v \in V$.

The total flow of evacuees leaving source s equals the total flow of the evacuees held at any vertex $v \in V$ over the time horizon T, i.e.,

$$\sum_{\theta=0}^{T} \sum_{e \in \delta^+(s)} f(e, \theta) = \sum_{v \in V} h_{v'}. \tag{2.5}$$

3. Solution Discussion

The modified preflow-push algorithm in [10] solves the maximum evacuation planning problem for two-terminal static case that allows holding of evacuees in temporary shelter at intermediate nodes. The procedure does not send the evacuees reaching once at intermediate nodes back again to the source, a dangerous place but may push back to any intermediate nodes. The preflow in each iteration is updated in the residual network N_f. For an arc $(v, w) \in N$ and $f(v, w) < c_{(v,w)}$, N_f contains the arc (v, w) with residual capacity $r_{(v,w)} = c_{(v,w)} - f(v, w)$ and if $(v, w) \in N$ and $f(v, w) > 0$, then N_f contains the arc (w, v) with residual capacity $r_{(w,v)} = f(v, w)$. The preflow is pushed from an active node $v \in V$ to its neighboring node $w \neq s$, $(v, w) \in N_f$ when the label at v and w satisfies $l(v) = l(w) + 1$. Otherwise, the label of v is increased as $l(v) := 1 + min\,\{l(w) : (v, w) \in N_f\}$. The label function $l : V \to Z^+ \cup \{0\}$ is defined as

$$l(v) \begin{cases} = |V| & \text{if } v = s, \\ \leq l(w) + 1 & \text{if } (v, w) \in N_f, \\ = 0 & \text{if } v = d. \end{cases}$$

The excess flow of evacuees at v is held at the temporary shelter $v^{'}$ if the push operation and the relabel operation are not applicable and even if $l(v) \leq l(s)$ does not satisfy after relabeling it. The procedure terminates with a maximum number of evacuees into the sink as well as into the possible temporary shelters at the intermediate places simultaneously and runs with polynomial time complexity. The exact solution procedures are given in Algorithm 3.1 and Algorithm 3.2.

Algorithm 3.1. Subroutine: Push-Relabel-Hold

Push: *For an arc* $(v, w) \in N_f$ *of active node* v *with* $l(v) = l(w)+1$*, push* $\delta = min\left\{e_v, c_{(v,w)}\right\}$ *flow units along* (v, w)*. Otherwise,*
Relabel: *For an active node* v*, set* $l(v) := 1 + min\left\{l(w) : (v, w) \in N_f\right\}$ *satisfying* $l(v) \leq l(s)$*. Otherwise,*
Hold: *Hold excess* $e_v > 0$ *at* v'*.*

Algorithm 3.2. Modified Pre-flow-Push Algorithm for Static Network

Input: *Network* $N = (V, E, c_e, s, d)$
Label Initialization:
For all $v \in V - \{s\}$*, set* $l(v)$ *to be the shortest path distance of* v *from* d *and set* $l(s) := n$*.*
Preflow Initialization:
Set $f(e) := c_e$ *for all* $e \in \delta^+(s)$ *and* $f(e) := 0$ *for remaining arcs.*
Subroutine Application:
Apply the subroutine PUSH-RELABEL-HOLD(v) for each active node $v \in V$*.*
Output: *Maximum static flow with intermediate hold on* N*.*

The intermediate nodes, in which the excess flow of evacuees might be held, may not be of equal importance with respect to the risk, the distance from the sink, the holding capacity etc. Solution to the problem with priority based intermediate nodes can be carried out by applying the modified algorithm repeatedly on the residual network N_f with k intermediate nodes ordered as $I_1,\ I_2,\ \ldots,\ I_k$ from lower priority to higher one after computing the maximum static flow f in N.

The modified algorithm also solves the problem for two-terminal network with dynamic case. One way of solving the problem is to apply the notion of time expanded network with necessary modification. The time-expanded network N^T, suggested in [5], of network N for time horizon T without holdover arcs is given by $N^T = (V^T, E^T)$ where V^T is the set of nodes $v(\theta)\ \forall\ v \in V\ \ \&\ \forall\ \theta \in \{0, 1, ..., T\}$ and E^T is the set of arcs $(v(\theta), w(\theta + \tau(v, w)))$ such that $v \neq w,\ v, w \in V\ \&\ \forall\ \theta \in \{0, 1, ..., T - \tau_e\}$.

For the modification, the arcs $(s(\theta), s(\theta{+}1))$ and $(d(\theta), d(\theta{+}1))$ for all $\theta \in \{0, 1, ..., T - 1\}$ with sufficient capacities are added to the set E^T. Moreover, each node $v(T); v \in V \setminus \{s\}$ is connected to $d(T)$ by an artificial arc with zero capacity. A small network N depicted

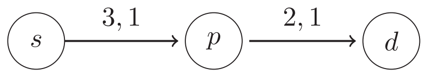


FIGURE 2. Network N.

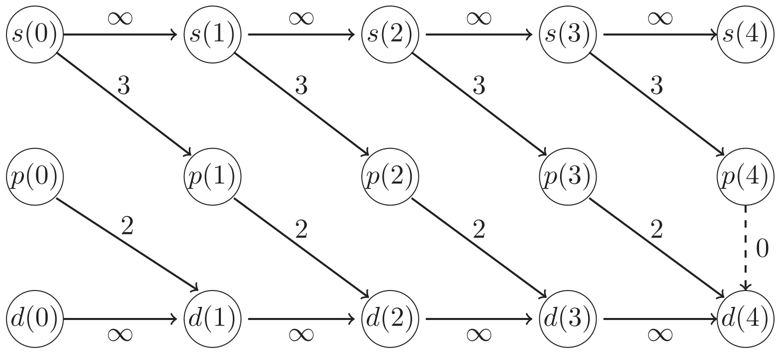


FIGURE 3. Time expanded network of the network depicted in Figure 2 for $T = 4$.

on Figure 2 has been expanded for time horizon $T = 4$ in Figure 3. The exact solution procedure that solves the maximum dynamic flow problem is given in Algorithm 3.3. The limitation associated to this procedure is that it leads to a pseudo-polynomial time complexity since it strongly depends on T.

Algorithm 3.3. ***Modified Preflow-Push Algorithm for Dynamic Network***

(1) *Given network* $N = (V, E, c_e, \tau_e, s, d, T)$.
(2) *Find* N^T *of* N.
(3) *Apply the Modified Preflow-Push Algorithm on* N^T *where* $s(0)$ *is the source and* $d(T)$ *is the sink.*
(4) *Get maximum dynamic flow with intermediate hold on* N.

---

4. Conclusion

The number of evacuees out of the source may exceed the number of evacuees entering into the sink in the evacuation planning problem modeled with no flow conservation. The solution procedure discussed in this paper is based on the preflow-push concept and maximizes the number of evacuees not only into the sink but into the possible intermediate places also. Our model assumes the sufficient holding capacities at intermediate nodes. However, the flow value that is held at intermediate node is also regulated by the residual capacity of the path from source to it. Investigation of more exact solution procedure leading to polynomial time complexity for the problem as well as modeling the problem with fixed holding capacity at the intermediate nodes would be the immediate research directions.

Acknowledgements: The second author would like to acknowledge University Grants Commission, Nepal for PhD Fellowship Award 2016.

References

[1] P. P. Bhandari, S. R. Khadka, Efficient Solution Approach to Maximum Flow Evacuation Planning Problem without Flow Conservation Aspect, *Journal of the Institute of Engineering*, Vol. 13, pp 108-116, 2017

[2] P. P. Bhandari, S. R. Khadka, Non-conserving Flow Aspect of Maximum Dynamic Flow Problem, *Journal of the Institute of Engineering*, Vol. 14, pp 107-114, 2018

[3] T.N. Dhamala, A Survey on Models and Algorithms for Discrete Evacuation Planning Network Problems. *Journal of Industrial and Management Optimization*, Vol. 11, pp 265–289, 2015

[4] L.R. Ford, D.R. Fulkerson, Maximal Flow through a Network, *Can. J. Math.*, Vol. 8, pp 399-404, 1956

[5] L.R. Ford, D.R. Fulkerson, Constructing Maximal Dynamic Flows from Static Networks, *Operations Research*, Vol. 6(3), pp 419-433, 1958

[6] L.R. Ford, D.R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962

[7] A.V. Goldberg, R.E. Tarjan, A New Approach to the Maximum-flow Problem, *Journal of ACM*, Vol. 35(4), pp 921-940, 1988

[8] S.R. Khadka, P.P. Bhandari, Dynamic Network Contraflow Evacuation Planning Problem with Continuous Time Approach, *International Journal of Operations Research*, Vol. 14(1), pp 27-34, 2017

[9] B. Kotnyek, An Annotated Overview of Dynamic Network Flows, *Technical Report, INRIA, Sophia Antipolis*, pp 1-28, 2004

[10] S.R. Khadka, P.P. Bhandari, An Efficient Solution Approach to Non-conservation Maximum Flow Evacuation Planning Problem, *Proceedings on SKIMA 2017, Colombo*, pp 333-335, 2017

[11] S. Kim, S. Shekhar, M. Min, Contraflow Transportation Network Reconfiguration for Evacuation Route Planning, *IEEE transactions on Knowledge and Data Engineering*, Vol. 20(8), pp 1115-1129, 2008

[12] U. Pyakurel, T.N. Dhamala, S. Dempe, Efficient Continuous Contraflow Algorithms for Evacuation Planning Problems, *Annals of Operations Research*, Vol. 254, pp 335–364, 2017

[13] S. Rebennack, A. Arulselvan, L. Elefteriadou, P.M. Pardalos, Complexity Analysis of Maximum Flow Problem with Arc Reversal, *Journal of Combinatorial Optimization*, Vol. 19, pp 200-216, 2010

Journal of Mathematics and Statistics

Original Research Paper

Lexicographically Maximum Dynamic Flow with Vertex Capacities

[1]Phanindra Prasad Bhandari, [1]Shree Ram Khadka, [2]Stefan Ruzika and [2]Luca E. Schäfer

[1]*Central Department of Mathematics, Tribhuvan University, Kirtipur, Kathmandu, Nepal*
[2]*Department of Mathematics, Technische Universitat Kaiserslautern, P.O. Box 3049, 67663 Kaiserslautern, Germany*

Article history
Received: 11-06-2020
Revised: 29-06-2020
Accepted: 23-07-2020

Corresponding Author:
Shree Ram Khadka
Central Department of Mathematics, Tribhuvan University, Kirtipur, Kathmandu, Nepal
Email: shreeramkhadka@gmail.com

Abstract: We consider an evacuation planning problem in the sense of computing a feasible dynamic flow lexicographically maximizing the amount of flow entering a set of terminals with respect to a given prioritization and given vertex capacities. We propose a polynomial time algorithm for the static version of the problem and a pseudo-polynomial time algorithm for the dynamic case. We show that by neglecting the vertex capacities, the dynamic version can be solved in polynomial time by using temporally repeated flows.

Keywords: Evacuation Planning, Disaster Management, Lexicographically Maximum Flows, Dynamic Flows, Vertex Capacities

Introduction

Mathematical optimization provides important tools for modeling, preparing and managing evacuation tasks, (Borrmann *et al*., 2012; Göttlich *et al*., 2011; Hamacher *et al*., 2011). The maximum flow evacuation planning problem asks for a flow which sends a maximum number of evacuees from a disastrous zone (source) to a safe zone (sink) within a given time horizon. In practical applications there might be relatively safe places apart from the sink. Hence, sending as many of the remaining evacuees as possible to these prioritized spots is desired. However, these spots might be constrained to some given vertex capacities, which restrict the amount of flow that can enter these vertices within the given time horizon.

Since the introduction of the maximum flow problem by (Ford and Fulkerson, 1956), the problem and its applications have been extensively studied in the literature, (Borradaile *et al*., 2017; Cherkassky and Goldberg, 1997; Goldberg and Tarjan, 1988). The lexicographically maximum flow problem has been investigated as a variant of the classical maximum flow problem, (Megiddo, 1974; 1977; Minieka, 1973). The authors showed that this problem can be solved in polynomial time. Dynamic extensions of these problems often provide important features for modeling real-world applications, e.g., in evacuation scenarios. Many dynamic network flow problems have been investigated in the context of evacuation planning problems, (Dhamala, 2015; Hamacher and Tjandra, 2001; Khadka and Bhandari, 2017; Pyakurel *et al*., 2017; Rebennack *et al*., 2010). To solve the maximum dynamic flow problem, a pseudo-polynomial time algorithm based on the construction of a time-expanded graph and a polynomial time algorithm based on temporally repeated flows with transit times on the arcs treated as cost coefficients have been investigated by (Ford and Fulkerson, 1958; 1962). Hoppe and Tardos (1994; 2000) study lexicographically maximum dynamic flows. They developed a polynomial time algorithm based on temporally repeated flows that lexicographically maximizes the flow leaving the terminals of an ordered terminal set consisting of sources and sinks. This is equivalent to lexicographically minimizing the flow entering the sinks in the given order.

In this study, we consider a maximum flow evacuation planning problem with a prioritized terminal set with fixed vertex capacities for each of these vertices. We aim to lexicographically maximize the amount of flow entering the vertices in the terminal set within a given time horizon with respect to the prioritization and the holding capacities. This problem is motivated by the situation encountered in evacuation scenarios: As many evacuees as possible are to be sent to safety. However, if sending evacuees to safety is not possible within a given time horizon, it is desirable to send as many evacuees to shelters of limited capacity, which are located within the evacuation zone. In contrast to the above mentioned models, we assume that the terminals have a fixed vertex capacity delimiting the amount of flow that can enter a vertex within a given time horizon. We provide a polynomial time algorithm for the static version of the problem and a pseudo-polynomial time algorithm for the dynamic case based on the construction of a time-expanded network. We show how to modify the lexicographically maximum dynamic flow algorithm (Hoppe and Tardos, 1994; 2000) to solve our problem in the case of neglecting the vertex capacities. The optimal routes identified by the procedure we propose in this study allows to send more evacuees out from risk zone to relatively safe places, besides maximum evacuees to the safe zone(sink), at least for some time during the period of response in emergency mitigation.

The remainder of this paper is structured as follows. At first, we formally introduce the maximum flow evacuation planning problem. Then we consider the problem in static and dynamic versions one after another, respectively. Finally, we conclude the paper with some further research objectives.

Problem Formulation

Let $G = (V, A)$ denote a directed graph with vertex set V and arc set A. Both, the set of vertices and the set of arcs are assumed to be finite and we set $n := |V|$ and $m := |A|$. We denote the source and the sink vertex by s and d, respectively. We assume that the graph G does not contain parallel arcs nor loops. Further, we assume that there are no arcs entering s and leaving t, respectively. By $\delta^-(v)$ and $\delta^+(v)$ we denote the set of arcs entering and leaving vertex $v \in V$, respectively. We assume a lower and upper arc capacity function l, u: $A \to \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ to be given, which bounds the number of flow units on each arc at each time step from below and from above. Most of the time, we set $l(a) = 0$ for all $a \in A$. Further, a transit time function t: $A \to \mathbb{N}_0$ specifies the time needed by a flow unit to traverse an arc. We assume a terminal set $S \subset V$ with $S := \{v_1,\ldots,v_k\}$ prioritized from higher to lower priority, i.e., $v_1 \succ v_2 \succ \cdots \succ v_k$, to be given, where $v_1 = d$. Further, we define a vertex capacity function k: $S \to \mathbb{N}_0$ delimiting the total number of flow units, which may be held in each of the vertices $v \in S$. We set $k(d) = \infty$ and $k(v)$ to be finite for all $v \in \mathscr{S} \setminus \{d\}$. In the following, we assume a time horizon $T \in \mathbb{N}$ to be given and treat time in a discrete manner, i.e., $\mathscr{T} := \{0, 1,\ldots,T\}$. Summing it up, we denote by $\mathscr{N} = (G, l, u, \tau, T, k)$ a dynamic network. If we aim to refer to a static network without transit times, we just write $\mathscr{N} = (G, l, u, k)$.

To this end, nonnegative flow variables f: $A \times \mathscr{T} \to \mathbb{N}_0$ specify the flow over time in the network $\mathscr{N}$. More precisely, $f(a, t)$ equals the number of flow units entering arc a at time step t. Further, flow that enters arc a at time t, reaches the end of arc a at time $t + \tau(a)$. The number of flow units entering arc a at time step t are assumed to be bounded by the capacity of an arc, i.e., $0 \le f(a, t) \le u(a)$ for all $a \in A$ and for all $t \in \mathscr{T}$. Moreover, $f(a, t)$ has to be equal to zero for all $t > T-\tau(a)$ and for all $a \in A$. The excess of a vertex $v \in V$ at time $t \in \mathscr{T}$ is defined as:

$$ex_f(v,t) := \sum_{a \in \delta^-(v)} \sum_{\xi=0}^{t-\tau(a)} f(a,\xi) - \sum_{a \in \delta^+(v)} \sum_{\xi=0}^{t} f(a,\xi).$$

Consequently, we need to ensure that $ex_f(v, T) \le k(v)$ for all $v \in \mathscr{S}$.

The objective function of the maximum flow evacuation planning problem asks to lexicographically maximize the vector $(ex_f(v_1, T),\ldots, ex_f(v_k, T))^\top$ such that $ex_f(v_i) \le k(v_i)$ for $i = 1,\ldots,k$. Note that $k(v_1) = k(d) = \infty$ and $v_i \in \mathscr{S}$ for $i = 1,\ldots,k$.

Static Version

Let $\mathscr{N} = (G, l, u, k)$ be a static network without transit times. Further, let $\mathscr{S} = \{v_1,\ldots,v_k\}$ with $d = v_1 \succ \ldots \succ v_k$ prioritized from higher to lower priority. Moreover, we assume that $l(a) = 0$ for all $a \in A$. The goal is to compute a lexicographically maximum flow in $\mathscr{N}$ satisfying the arc capacities for all arcs and the vertex capacities for all $v \in \mathscr{S}$. This is achieved by iterative maximum flow computations in a transformed network as described in the following. First, the network $\mathscr{N}$ with vertex capacities is transformed into a network $\mathscr{N}'$ without vertex capacities. We introduce an artificial vertex v_i' for each vertex $v_i \in \mathscr{S}$. We call the artificial vertex $v_1' := d'$ the supersink. Then, the vertices v_i and v_i' are connected by an artificial arc (v_i, v_i') with $u(v_i, v_i') = k(v_i)$. Moreover, each vertex v_i' is linked to the supersink d' by introducing an artificial arc (v_i', d') having zero arc capacity. Only the artificial arc (d, d') gets infinite arc capacity. Further, every arc (original and artificial) gets a lower arc capacity of zero, i.e., $l(a) = 0$ for all a. Doing this, the network $\mathscr{N} = (G, l, u, k)$ is transformed into the network $\mathscr{N}' = (G', l, u)$ with $G' = (V', A')$, Fig. 1.

Algorithm 1 works as follows. Since $v_1 = d$ is the vertex with the highest priority, a maximum flow from source s to sink d is computed by applying a maximum flow algorithm in the transformed network $\mathscr{N}'$. Next, lower and upper capacities are updated in the following manner: The lower capacities are set to zero, whereas the upper capacities remain the same; only the arc (d, d') gets a lower and upper capacity equal to the value of the previously computed maximum flow from s to d'. Next, we aim to maximize the flow from s to the vertex v_i' with the next highest priority.

DOI: 10.3844/jmssp.2020.142.147

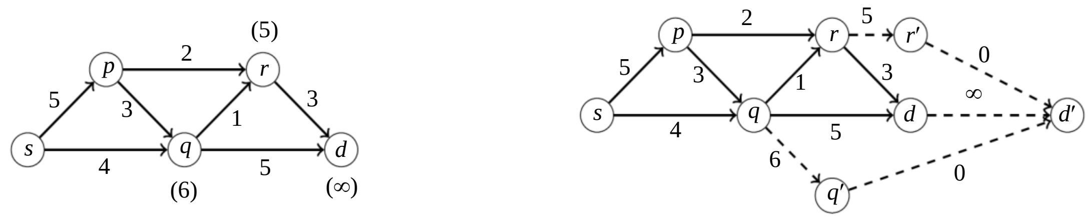


Fig. 1: Network $\mathcal{N}$ with $\mathcal{S}$ = {d, r, q}, where the numbers on the arcs refer to the arc capacities and the numbers in parenthesis above and below the vertices refer to the vertex capacities (left). Transformed network $\mathcal{N}'$ (right).

Algorithm 1 A lexicographic maximum static flow algorithm

Input: A static network $\mathcal{N}$ = (G, l, u, k), $\mathcal{S}$ = $\{v_1,\ldots,v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$

Output: A lexicographically maximum flow satisfying the vertex capacities for all vertices in $\mathcal{S}$

1: $d' \leftarrow$ super sink, $V \leftarrow V \cup \{d'\}$, $A \leftarrow A \cup \{(v_1, d')\}$ with $u(v_1, d') = \infty$
2: Compute maximum s-d'-flow in G and let f^* be the optimal value
3: $l(v_1, d') \leftarrow f^*$ and $u(v_1, d') \leftarrow f^*$
4: *for i* = 2,…,*k do*
5: $\quad V \leftarrow V \cup \{v_i'\}, A \leftarrow A \cup \{(v_i, v_i'), (v_i', d')\}$
6: $\quad u(v_i, v_i') \leftarrow k(v_i), l(v_i, v_i') \leftarrow 0$
7: $\quad u(v_i', d') \leftarrow \infty$ and $l(v_i', d') \leftarrow 0$
8: $\quad$ Compute maximum s-d'-flow in $\mathcal{N}$ and let f^* be the optimal value
9: $\quad$ Let $f^*(v_i', d')$ be the flow value on arc (v_i', d') w.r.t. f^*
10: $\quad u(v_i', d') \leftarrow f^*(v_i', d')$ and $l(v_i', d') \leftarrow f^*(v_i', d')$

This is achieved by setting $u(v_i', d')$ to infinity and again computing a maximum flow from s to d' with lower and upper arc capacities. Note that we do not have to find a feasible flow in the transformed network, since the maximum flow computed in the previous iteration is already a feasible flow in the modified network. Further, note that due to the lower and upper capacities on (d, d'), it is ensured that the previously computed maximum flow value from s to d' remains the same. This procedure of computing maximum s-v_i-flows based on previously computed flows is iteratively repeated for all vertices $v_i \in \mathcal{S}$.

Theorem 2.1

Given a network $\mathcal{N}$ = (G, l, u, k), source s and terminal set $\mathcal{S} = \{v_1,\ldots,v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$ and $l(a) = 0$ for all $a \in A$. Then, Algorithm 1 computes a lexicographical maximum flow in $\mathcal{N}$ in polynomial time.

Proof

Obviously, the maximum flow value f^* from s to d' (see line 2) is equal to the maximum flow value in the original network $\mathcal{N}$ from s to d. Let v_j be the vertex with the next highest priority. One can see that the maximum s-d'-flow remains feasible when setting $u(v_1, d')$ and $l(v_1, d')$ to be equal to f^* and introducing arc (v_j', d') with $u(v_j', d') = \infty$. If we compute again a maximum s-d'-flow, then the flow value on arc (v_1, d') is equal to the maximum s-d'-flow computed in the previous iteration, whereas the flow value on arc (v_j', d') is equal to the maximal possible flow that can be sent to v_j' among all maximal s-d'-flows. Repeating this argument, we get a lexicographical maximum s-v_i-flow for all $v_i \in \mathcal{S}$. The polynomial running time follows from the fact that at most $k \le n$ maximum flow problems are to be solved, which can be done in polynomial time.

Dynamic Version

Time-Expanded Network

Let $\mathcal{N}$ = (G, l, u, τ, T, k) be a dynamic network. Again, let $\mathcal{S} = \{v_1,\ldots,v_k\}$ with $v_1 \succ \ldots \succ v_k$ be prioritized from higher to lower priority and $l(a) = 0$ for all $a \in A$. Further, without loss of generality we assume that vertices in $\mathcal{S}$ have no outgoing arcs. We transform the dynamic network $\mathcal{N}$ into a time-expanded network $\mathcal{N}^T = (V^T, A^T, l', u')$ in the following way:

- $V^T := \{v_t \mid t = 0,1,\ldots,T\}$,
- $A^T := \{(v_t, w_{t'}) \mid (v = w \wedge t' = t+1) \vee ((v, w) \in A \wedge \tau(v, w) = t'-t)\}$,
- $l'(a) = 0$ for all $a \in A^T$,
- $u'(v_t, w_{t'}) = u(v, w)$ for all $(v, w) \in A$ and $u'(v_t, v_{t+1}) = \infty$ for all $v \in V$.

Algorithm 2 A discrete dynamic lexicographic maximum flow algorithm

Input: A dynamic network $\mathcal{N} = (G, l, u, \tau, T, k)$, $\mathcal{S} = \{v_1,\ldots,v_k\}$ with $d = v_1 \succ \cdots \succ v_k$, $l(a) = 0$ for all $a \in A$

Output: A discrete dynamic lexicographically maximum flow satisfying the vertex capacities for all vertices in $\mathcal{S}$

1: Create the time-expanded network as described above
2: **for all** $v \in \mathcal{S}$ **do**
3: $V^T \leftarrow V^T \cup \{v'\}$, $A^T \leftarrow A^T \cup \{(v_T, v')\}$ $\quad \triangleright\ v' := v'_T$
4: $u'(v_T, v') \leftarrow k(v_i)$, $l'(v_T, v') \leftarrow 0$
5: Compute a static maximum s_0-d'-flow and let f^* be the optimal value
6: $l'(d_T, d') \leftarrow f^*$ and $u'(d_T, d') \leftarrow f^*$
7: **for** $i = 2,\ldots,k$ **do**
8: $A^T \leftarrow A^T \cup \{(v'_i, d')\}$
9: $u'(v'_i, d') \leftarrow \infty$, $l'(v'_i, d') \leftarrow 0$
10: Compute a maximum s_0-d'-flow and let f^* be the optimal value
11: Let $f^*(v'_i, d')$ be the flow value on arc (v'_i, d') w.r.t. f^*
12: $u'(v'_i, d') \leftarrow f^*(v'_i, d')$ and $l'(v'_i, d') \leftarrow f^*(v'_i, d')$

Next, as described in the static version, we introduce a vertex $v'_i := v'_{i_T}$ for all v_{i_T} with $v_i \in \mathcal{S}$. We connect vertices v_{i_T} and v'_i by arcs with upper capacity $k(v_i)$ for all $v_i \in \mathcal{S}$ (and zero transit time, since v_{i_T} and v'_i are on the same time level). Obviously, a static maximum s_0-d'-flow, i.e., a s_0-v'_1 -flow, in the time-expanded network corresponds to a discrete dynamic s-d-flow in $\mathcal{N}$. After computing the static maximum s_0-d'-flow, we set $l'(d_T, d')$ as well as $u'(d_T, d')$ to the value of the maximum s_0-d'-flow. Afterwards, we introduce an arc from v'_i to d' of infinite capacity, where v_i is the vertex with the next highest priority. Then, we compute again a static maximum s_0-d'-flow in the time-expanded network. This procedure is iteratively repeated (similar to Algorithm 1) such that we obtain a discrete dynamic lexicographically maximum flow, see Algorithm 2.

Corollary 3.1

Given a network $\mathcal{N} = (G, l, u, \tau, T, k)$, source s and terminal set $\mathcal{S} = \{v_1,\ldots,v_k\} \subset V$ with $d = v_1 \succ \cdots \succ v_k$ and $l(a) = 0$ for all $a \in A$. Then, Algorithm 2 computes a discrete dynamic lexicographical maximum flow in $\mathcal{N}$ in pseudo-polynomial time.

Proof

The correctness follows from Theorem 2.1. The pseudo-polynomial running time follows from the fact that the size of the time-expanded graph is pseudo-polynomial in the input size.

Temporally Repeated Flows

An optimal solution to minimum cost circulation, flow problem obtained by interpreting transit times as cost coefficients for each arc $a \in A$, can be transformed into a maximal discrete dynamic flow for single-source-single-sink case using the notion of Temporally Repeated Flows (TRFs) (Ford and Fulkerson, 1958). This technique computes feasible optimal flow for $v_1 = d$ as sink for the problem. However, while considering remaining vertices $v_i \in \mathcal{S}$ as the sink, flow computed by TRFs may exceed fixed vertex capacities. Moreover, TRFs may not induce optimal flows for these vertices as sinks due to non-uniqueness of path decomposition carried out for TRFs. These hurdles occur due to the fixed vertex capacities at intermediate vertices. In the following we consider the problem on network without vertex capacities.

Let $\mathcal{N} = (G, l, u, \tau, T)$ be a dynamic network without vertex capacities. Let $\mathcal{S} = \{v_1,\ldots,v_k\}$ be a terminal set with $v_1 \succ \cdots \succ v_k$ prioritized from higher to lower priority and $l(a) = 0$ for all $a \in A$. We aim to solve our problem in polynomial time without vertex capacities by using the lexicographically maximum dynamic flow algorithm proposed in (Hoppe and Tardos, 2000). Their algorithm is summarized in Algorithm 3. Note that $\mathcal{N}^i_{g^{i+1}}$ refers to the residual dynamic network with respect to flow g^{i+1} with vertex set V and arc set A^i.

Algorithm 3 lexicographically maximizes the amount of flow leaving the terminals in $\mathcal{S}$ in the given order, i.e., the algorithm lexicographically maximizes the vector $(-ex_f(v_1, T), \ldots, -ex_f(v_k, T))^\top$. However, we aim at lexicographically maximizing $(ex_f(v_1, T), \ldots, ex_f(v_k, T))^\top$. Therefore, we adapt our problem in the following way such that we can use Algorithm 3 to solve it:

1) $V \leftarrow V \cup \{v'_i\}$, $A \leftarrow A \cup \{(v_i, v'_i)\}$ for all $v_i \in \mathcal{S}$
2) $u(v_i, v'_i) \leftarrow \infty$ and $\tau(v_i, v'_i) \leftarrow 0$
3) $\mathcal{S} = \{v'_1, \ldots, v'_k, s\}$ with $\mathcal{S}^+ \leftarrow \mathcal{S} \setminus \{s\}$ and $\mathcal{S}^- \leftarrow \{s\}$
4) Take the inverse network of $\mathcal{N}$, i.e., $\mathcal{N}^{inv}$, where all arcs are reversed
5) Apply Algorithm 3 on $\mathcal{N}^{inv}$

Algorithm 3 Lexicographically maximum dynamic flow algorithm (Hoppe and Tardos, 2000)

Input: A dynamic network $\mathcal{N} = (G, u, \tau, T)$, $\mathcal{S} = \{v_1,\ldots,v_k\} = \mathcal{S}^+ \dot\cup \mathcal{S}^-$ with $v_1 \succ \cdots \succ v_k$, where $\mathcal{S}^+$ and $\mathcal{S}^-$ refer to the sources and sinks of $\mathcal{S}$, respectively

Output: A lexicographically maximum dynamic flow

$V \leftarrow V \cup \{s^*\}$ ▷ Introduce super source s^*
$A^{k+1} \leftarrow A \cup \{(s^*, s) \mid s \in \mathcal{S}^*\}$, $u(s^*, s) \leftarrow \infty$, $t(s^*, s) \leftarrow 0$
$g^{k+1} \leftarrow 0$ ▷ zero flow
$\Gamma^{k+1} \leftarrow \emptyset$ ▷ path decomposition
for $i = k, \ldots, 1$ **do**
    $A^i \leftarrow A^{i+1}$
    if $v_i \in \mathcal{S}^-$ **then**
        $A^i \leftarrow A^i \cup \{(v_i, s^*)\}$, $u(v_i, s^*) \leftarrow \infty$, $t(v_i, s^*) \leftarrow -(T+1)$
        $f^i \leftarrow$ min cost circulation in $\mathcal{N}^i_{g^{i+1}}$ with t as arc costs
    if $v_i \in \mathcal{S}^+$ **then**
        $A^i \leftarrow A^i \setminus \{(s^*, v_i)\}$
        $f^i \leftarrow$ min. cost max. s^*-v_i-flow in $\mathcal{N}^i_{g^{i+1}}$ with τ as arc costs
    $g^i \leftarrow g^{i+1} + f^i$
    $P^i \leftarrow$ path decomposition of f^i
    $\Gamma^i \ \Gamma^{i+1} \cup P^i$
return Γ^1

This procedure yields a dynamic flow in $\mathcal{N}^{inv}$ lexicographically maximizing the amount of flow leaving the terminals in $\mathcal{S}$ in the given order. By translating the obtained dynamic flow back to the network $\mathcal{N}$, we obtain the desired dynamic flow lexicographically maximizing the amount of flow entering the terminals in the given order. The correctness of the procedure follows immediately from the correctness of the lexicographically maximum dynamic flow algorithm, see Hoppe and Tardos (2000). Since the overhead of this procedure is determined by the computation of the k minimum cost flows, the polynomial runtime follows.

Conclusion

In this article, we have introduced a maximum flow evacuation planning problem, where one aims to lexicographically maximize the amount of flow entering the vertices of a given prioritized terminal set with respect to vertex capacities. We showed how to solve this problem in polynomial time for the static case and in pseudo-polynomial time for the dynamic case in the time-expanded network. By neglecting the vertex capacities, we provided a procedure to solve that problem in polynomial time in a dynamic network. This work identifies optimal routes to the prioritized vertices besides the safe zone (sink). This allows to send more evacuees out from the disastrous zone (source) to the relatively safe places at least for some time during the period of response in emergency mitigation.

The main shortcoming of this work is one cannot repeatedly send the evacuees at those vertices where vertex capacity is fixed. In the future, it would be interesting to see how to solve that problem in the dynamic case by using temporally repeated flows and satisfying given vertex capacities.

Acknowledgment

First author would like to thank University Grants Commission, Nepal for PhD Fellowship Award 2016. First and second authors are also grateful to GraThO project between TU Kaiserslautern, Germany; TU, Nepal and MSU, Philippines supported by DAAD for providing a suitable atmosphere to conduct this research work. This work was partially supported by the Bundesministerium fur Bildung und Forschung (BMBF) under Grant No. 13N14561. Authors are grateful to anonymous referees for their insightful comments that significantly improved the paper.

Author's Contributions

Phanindra Prasad Bhandari: Prepared the initial manuscript with solution in static and dynamic over the time expanded graph.

Shree Ram Khadka: Initiated the problem, developed the model and polished.

Stefan Ruzika: Improved the model and solution procedure and polished.

Luca E. Schäfer: Contributed in dynamic case with temporally repeated approach and improved the manuscript.

Ethics

There is no non-ethical issues involved in the article. It is original and contains unpublished materials.

References

Borradaile, G., Klein, P. N., Mozes, S., Nussbaum, Y., & Wulff-Nilsen, C. (2017). Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. SIAM Journal on Computing, 46(4), 1280-1303.

Borrmann, A., Kneidl, A., Köster, G., Ruzika, S., & Thiemann, M. (2012). Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models. Safety science, 50(8), 1695-1703.

Cherkassky, B. V., & Goldberg, A. V. (1997). On implementing the push-relabel method for the maximum flow problem. Algorithmica, 19(4), 390-410.

Dhamala, T. N. (2015). A survey on models and algorithms for discrete evacuation planning network problems. Journal of Industrial & Management Optimization, 11(1), 265.

Ford, L. R., & Fulkerson, D. R. (1956). «Maximal Flow through a Network», Canadian Journal of Mathematics.

Ford, L. R., & Fulkerson, D. R. (1958). Constructing maximal dynamic flows from static flows. Operations research, 6(3), 419-433.

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

Goldberg, A. V., & Tarjan, R. E. (1988). A new approach to the maximum-flow problem. Journal of the ACM (JACM), 35(4), 921-940.

Göttlich, S., Kühn, S., Ohst, J. P., Ruzika, S., & Thiemann, M. (2011). Evacuation dynamics influenced by spreading hazardous material. Networks & Heterogeneous Media, 6(3), 443.

Hamacher, H. W., Heller, S., Klein, W., Köster, G., & Ruzika, S. (2011). A sandwich approach for evacuation time bounds. In Pedestrian and Evacuation Dynamics (pp. 503-513). Springer, Boston, MA.

Hamacher, H. W., & Tjandra, S. A. (2001). Mathematical modelling of evacuation problems: A state of art.

Hoppe, B., & Tardos, É. (1994, January). Polynomial Time Algorithms for Some Evacuation Problems. In SODA (Vol. 94, pp. 433-441).

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

Khadka, S. R., & Bhandari, P. P. (2017). Dynamic network contraflow evacuation planning problem with continuous time approach. International Journal of Operations Research, 14(1), 27-34.

Megiddo, N. (1974). Optimal flows in networks with multiple sources and sinks. Mathematical Programming, 7(1), 97-107.

Megiddo, N. (1977). A good algorithm for lexicographically optimal flows in multi-terminal networks. American Mathematical Society, 83(3).

Minieka, E. (1973). Maximal, lexicographic and dynamic network flows. Operations Research, 21(2):517–527.

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.

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.

American Journal of Applied Mathematics
2020; 8(4): 230-235
http://www.sciencepublishinggroup.com/j/ajam
doi: 10.11648/j.ajam.20200804.18
ISSN: 2330-0043 (Print); ISSN: 2330-006X (Online)

Evacuation Contraflow Problems with Not Necessarily Equal Transit Time on Anti-parallel Arcs

Phanindra Prasad Bhandari*, Shree Ram Khadka

Central Department of Mathematics, Tribhuvan University, Kirtipur, Kathmandu, Nepal

Email address:
phanindra.maths@gmail.com (P. P. Bhandari), shree.khadka@cdmath.tu.edu.np (S. R. Khadka)
*Corresponding author

To cite this article:
Phanindra Prasad Bhandari, Shree Ram Khadka. Evacuation Contraflow Problems with Not Necessarily Equal Transit Time on Anti-parallel Arcs. *American Journal of Applied Mathematics*. Vol. 8, No. 4, 2020, pp. 230-235. doi: 10.11648/j.ajam.20200804.18

Received: July 27, 2020; **Accepted**: August 10, 2020; **Published**: August 17, 2020

Abstract: An evacuation planning problem provides a plan for existing road topology that sends maximum number of evacuees from risk zone to the safe destination in minimum time period during disasters. The problems with different road network attributes have been studied, and solutions have been proposed in literature. Evacuation planning problems with network contraflow approach, reversing the direction of traffic flow on lanes, with the same transit time on anti-parallel arcs have also been extensively studied. The approach, due to its lane-direction reversal property, can be taken as a potential remedy to mitigate congestion and reduce casualties during emergencies. In this paper, we propose a mathematical optimization contraflow model for the evacuation problem with the case where there may exist different transit time on anti-parallel arcs. We also propose analytical solutions to a few variants of problems, such as maximum dynamic contraflow problem and earliest arrival contraflow problem in which arc reversal capability is allowed only once at time zero. We extend the solution to solve the problems with continuous time settings by applying the natural relation between discrete time flows and continuous time flows. The solution procedures are based on application of temporally repeated flows (TRFs) on modified network, and they solve the problems optimally in strongly polynomial time.

Keywords: Network Flow, Contraflow, TTSP Network, Evacuation Planning Problem, Disaster Management

1. Introduction

An evacuation planning problem, important notion during the response phase of disaster management, attempts to find an optimal evacuation plan with a realistic flow model where each evacuee is supposed to be evacuated in a minimal time period from a risky site (source) to a safe site (sink). An efficient evacuation plan minimizes human casualties and their property during natural and human-created disasters, and also applicable in mitigation of rush-hour traffic in the crowded urban area.

The reversibility of direction of traffic flow in one or more lanes of roadways for fixed time period is termed as contraflow. The contraflow approach reconfigures the network identifying ideal direction and reallocating available capacity for each arc to improve the flow egress time and/or improve the number of flow units from source to destination. The approach, due to its lane-direction reversal property, can be taken as a potential remedy to mitigate congestion during emergencies by increasing outbound evacuation route capacity. It significantly reduces the total evacuation time and/or increase the number of evacuees sent from risk zone to safety. Studies show that reversing one lane of a four-lane dual highway increases the evacuation road capacity by approximately 30% and reversing all the inbound lanes, it increases by 67% [1].

Despite the long history of studies of evacuation problems with contraflow approach, there is limited implementation in real emergency evacuations due to difficulty in using commonly employed methods to duplicate traffic conditions of real contraflow lane during an emergency [1]. However, they have been adapted for evacuating some major metropolitan regions threatened by disasters. It was first applied during Hurricane Floyd in the United States in 1999 with mixed, though overall positive, results [2]. Contraflow was also implemented during hurricanes Katrina and Rita in the United States in 2005. However, it was criticized as unplanned contraflow orders and as failure to use contraflow lanes [3]. Contraflow approach is primarily important for emergency evacuations, nonetheless, its applications are not limited to these. This is commonly used for accommodating directionally imbalanced traffic associated with daily commuter in big cities as well as consequences due to religious gathering, concerts, tournaments, etc.

Maximum dynamic flow problem that sends the maximum amount of flow from the source to the sink within the given time horizon was due to Ford and Fulkerson [4]. Various applications of this problem including evacuation planning problems are considered in the literature, e.g., [5-7]. Evacuation problems that allow evacuees to be held at temporary shelters at intermediate spots have also been studied in [8-10].

The first mathematical optimization model for contraflow problem is due to Rebennack et al. [11]. They have investigated analytical solution of maximum contraflow problem with polynomial time complexity for both static and dynamic networks. Their solution idea is based on transformation of given network into one at which the existing algorithms are applicable. There is extensive study of dynamic network flow problems with continuous time setting, e.g., [12-17]. The continuous time dynamic network contraflow problems have been considered in [18] and [19]. The earliest arrival flow (EAF) problem that ask to maximize flow into the sink at each time points within the time horizon have also been considered widely in the literature, e.g., [20-25]. The EAF problem for two terminal series parallel (TTSP) network without and with contraflow approach have been studied and proposed polynomial time solutions in [26] and [27], respectively. Contraflow approach has been incorporated in network flow model to study facility location problem in [28] and the notion of abstract flow has been applied to network contraflow problems in [29]. The partial contraflow approach over the abstract network setting has been introduced in [30].

Time parameter plays a vital role in designing evacuation planning models. Therefore, it is important to be careful about its nature: discrete or continuous, adapted in the model. Optimization contraflow models developed so far are based on equal transit time settings on anti-parallel arcs and these models do not allow multiple arcs of different transit time. We call the two directed arcs 'anti-parallel' if they join the same pair of nodes, but in opposite directions. It is crucial, in case of uneven road architecture, for example, to take contraflow models on networks with not necessarily equal transit time on anti-parallel arcs into account for preparing evacuation tasks.

In this paper, we propose mathematical optimization contraflow model with assumption where the transit time on anti-parallel arcs may have different values. Discrete as well as continuous aspect of transit time are considered in the model, whereas capacities and transit time on arc are time independent and behave symmetrically during the reversal of direction of arcs.

Remaining part of the paper is organized as follows. Mathematical formulation of the contraflow problems are given in Section 2. Section 3 contains solution of maximum dynamic contraflow problem with discrete time setting and with continuous time setting in Subsection 3.1 and 3.2, respectively, and that of earliest arrival contraflow problem in Subsection 3.3. Section 4 concludes the paper.

2. Problem Formulation

Consider an evacuation network $N = (V, E, c(e), \tau(e), T)$ where V is the set of nodes v denoting the crossing of routes from dangerous place, the source s, to safer place, the sink d and E is the set of route segments, arc $e = (v, w)$ joining any two different nodes $v, w \in V$. Let $c: E \to Z^{\geq 0}$ be a capacity function denoting the upper bound for flow units to pass the arc at a time slot and $\tau: E \to R^{>0}$ be the transit time function denoting the time required for a flow unit to travel the arc. Moreover, we assume that the network N with not necessarily equal transit time on anti-parallel arcs. However, the transit time behave symmetrically during contraflow process. The total evacuation time period is denoted by T and we call it time horizon. An evacuation network has been depicted in Figure 1.

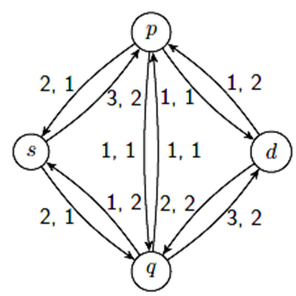


Figure 1. *An evacuation network N with source node s and sink node d. First and second numbers next to each arc are capacities and transit time, respectively.*

The flow of evacuees, say f, on the network N defined as $f: E \times [0, T) \to R^{\geq 0}$ satisfies the following conditions: Flow units travelling along arc e cannot exceed the arc capacity $c(e)$ for any time within given time horizon T. That is,

$$0 \leq f(e, \theta) \leq c(e) \; \forall \, e \in E \text{ and } \forall \, \theta \in [0, T). \qquad (1)$$

Flow units that enter into node v for all $v \in V \backslash \{s, d\}$ must exit from it within given time horizon. That is,

$$\sum_{e \in \delta^-(v)} \int_0^{T-\tau(e)} f(e, \theta) d\theta = \sum_{e \in \delta^+(v)} \int_0^{T} f(e, \theta) d\theta,$$

$$\forall \, v \in V \backslash \{s, d\}. \qquad (2)$$

where $\delta^-(v)$ and $\delta^+(v)$ denote for the set of arcs entering into the node v and leaving from it, respectively.

A dynamic $s-d$-flow on N that satisfies capacity constraints (1) is a feasible $s-d$-flow. For a dynamic network $N = (V, E, c(e), \tau(e), s, d, T)$, the objective of maximum dynamic $s-d$-contraflow problem with continuous time setting is to maximize a net feasible continuous dynamic flow, say f_c, from s to d within the given time horizon T, if the direction of the arcs on N can be reversed. The net flow f_c is given by

$$\boldsymbol{f_c} \coloneqq \sum_{e\in\delta^+(s)} \int_0^T f(e,\theta)d\theta - \sum_{e\in\delta^-(s)} \int_0^T f(e,\theta)d\theta$$

$$= \sum_{e\in\delta^-(d)} \int_{\theta=0}^{T} f(e,\theta)d\theta - \sum_{e\in+(d)} \int_{\theta=0}^{T} f(e,\theta)d\theta \quad (3)$$

If one wishes to send packets of flow units at discrete time points into the arcs instead of sending flow at continuous flow rates, the time horizon T is to be discretized into the time steps $\{0,1,\dots,T\}$. In discrete time flow model, the flow units sent into an arc $e=(v,w)$ at time θ totally reach the target node w at time $\theta+\tau(e)$, for $\tau: E \to Z^{>0}$. In discrete time setting, the flow function f defined as $f: E\times\{0,1,\dots,T\} \to Z^{\geq 0}$ satisfies the capacity constraint in the form:

$$0 \leq f(e,\theta) \leq c(e)\ \forall\ e \in E\ \&\ \forall\ \theta \in \{0,1,\dots,T\} \quad (4)$$

and the flow conservation constraint in the form:

$$\sum_{e\in\delta^-(v)} \sum_{\theta=0}^{T-\tau(e)} f(e,\theta) = \sum_{e\in\delta^+(v)} \sum_{\theta=0}^{T} f(e,\theta) \quad (5)$$

A dynamic $s-d$-flow satisfying capacity constraints (4) is a feasible $s-d$-flow for discrete time setting. For a dynamic network N and T being discretized, the maximum dynamic $s-d$ contraflow problem maximizes the net feasible discrete dynamic flow, say $\boldsymbol{f_d}$, from s to d within the given time horizon T, if the direction of the arcs on N can be reversed. The net flow $\boldsymbol{f_d}$ is given by

$$\boldsymbol{f_d} \coloneqq \sum_{e\in\delta^+(s)} \sum_{\theta=0}^{T} f(e,\theta) - \sum_{e\in\delta^-(s)} \sum_{\theta=0}^{T} f(e,\theta)$$

$$= \sum_{e\in\delta^-(d)} \sum_{\theta=0}^{T} f(e,\theta) - \sum_{e\in\delta^+(d)} \sum_{\theta=0}^{T} f(e\ \theta). \quad (6)$$

A maximum dynamic $s-d$-contraflow problem is also known as a *maximum dynamic contraflow (MDCF) problem* for single-source-single-sink network. Obviously, the flow value before and after the time horizon T is zero and all flow units leave the network within it in both discrete and continuous cases.

For given network $N=(V,E,c(e),\tau(e),s,d,T)$, *earliest arrival contraflow (EACF) problem*, if the direction of the arcs on N are allowed to reverse, maximizes the feasible net flow from s to d at each time points $\theta \in [0,T)$.

3. Solution Discussion

Authors in [11] studied maximum dynamic contraflow problem with equal transit time on anti-parallel arcs. Their model is with discrete time setting and arc reversal capability has been allowed only once at time zero. In this section, we consider maximum dynamic contraflow problems on dynamic networks in which there may be unequal transit time on anti-parallel arcs with discrete as well as continuous time settings in which arc reversal capability is allowed only once at time zero. We also discuss the solution procedure for earliest arrival contraflow problem with these settings for two terminal series parallel (TTSP) network.

3.1. Maximum Dynamic Contraflow with Discrete Time Setting

Consider the discrete time maximum dynamic contraflow (DT-MDCF) problem on network $N=(V,E,c(e),\tau(e),s,d,T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs. In the following, we propose a solution procedure to this problem when the arc reversibility is allowed only once at time zero.

Algorithm 1: Algorithm DT-MDCF

1. Given network $N=(V,E,c(e),\tau(e),s,d,T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs.
2. Transform network N into $N^+=(V^+, E^+, c(e^+), \tau(e^+), s, d, T)$ where $V^+=V\cup V'$ such that $V'=\{\overrightarrow{vw}: (v,w)\in E\}$ and $E^+=\{(v,\overrightarrow{vw}), (\overrightarrow{vw},w): v,w \in V, \overrightarrow{vw} \in V'\ \&\ (v,w)\in E\}$ with capacities $c(v,\overrightarrow{vw})=c(v,w)$, $c(\overrightarrow{vw},w)=\infty$ and transit time $\tau(v,\overrightarrow{vw})=\tau(v,w)$, $\tau(\overrightarrow{vw},w)=0$.
3. Transform network N^+ into its auxiliary network $\widetilde{N^+}$ as in [11].
4. Compute the discrete dynamic, temporally repeated flow on network $\tilde{N}$ for time horizon T.
5. Perform the flow decomposition into chain and cycle flows of the maximum flow obtained from step-4 and remove all cycle flows.
6. Arc $\overleftarrow{e} \in E$ is reversed if and only if the flow along arc $\overrightarrow{e} \in E$ is greater than c(e) or if there is non-negative flow along arc $e \notin E$.
7. Get discrete time maximum dynamic contraflow on N for time horizon T.

The procedure (cf. Algorithm 1) is based on the network transformation. In contrast to the case of equal transit time on the arcs, addition of capacities of anti-parallel arcs, while constructing the auxiliary network, is no longer possible in the case of unequal transit time. We propose an alternative, a more general, way of constructing the auxiliary network for the latter case. Network N with unequal transit time and capacities on the arcs is transformed into new network by introducing an artificial node for each arc that separates it into two different arcs. Each arc on N is split into two arcs: real arc and artificial arc, in the auxiliary network N^+. Artificial arcs have infinite capacities and zero transit time, whereas the real arcs have the original arc capacities and transit time. We denote artificial node that splits arc $(v,w)\in E$ by $\overrightarrow{vw}$, see Figure 2. Then the solution procedures that solves the MDCF problem with equal transit time on anti-parallel arcs, given in [11], is applicable on transformed network N^+. Their algorithm is based on reduction of given network N into its auxiliary network $\tilde{N}$. We denote $\overrightarrow{e}\in E$ for an arc (v,w) in which the flow unit is sent from the node v to the node w and $\overleftarrow{e}\in E$ for an arc (w,v) in which the flow unit is sent from the node w to the node v for all $v,w\in V$. Replacement of $\overrightarrow{e}$ by $\overleftarrow{e}$ is known as the arc reversal. For network N with equal transit time on anti-parallel arcs, its auxiliary network is $\tilde{N}=(V,\tilde{E},c(\tilde{e}),\tau(\tilde{e}),s,d,T)$ where

$\widetilde{E} = \{\tilde{e} = \vec{e} \text{ or } \overleftarrow{e}\}$ with $c(\tilde{e}) = c(\vec{e}) + c(\overleftarrow{e})$, and $\tau(\tilde{e}) = \tau(e)$ if $e \in E$ and $\tau(\tilde{e}) = \tau(\overleftarrow{e})$ otherwise. Any known technique can be applied to solve the maximum dynamic flow problem on $\widetilde{N}$, which ultimately, solves the original maximum dynamic flow problem with contraflow approach.

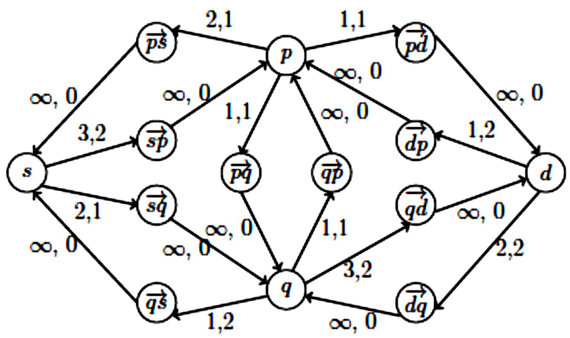


Figure 2. *Transformed network N^+ of the network N depicted in Figure 1.*

An optimal solution to minimum cost circulation (MCC) problem can be turned into a maximal discrete dynamic flow using the notion of temporally repeated flows (TRFs), [4]. The TRF is obtained by sending as much flow as possible along each path from source to sink at time zero and continue on them as long as there is enough time left within time horizon T to arrive at the sink d. To set the MCC problem on N, interpret the transit time $\tau(e)$ as cost coefficients for each arc $e \in E$. Also, it is desired to maximize dynamic flow on N so that the cost of circulation is minimum and that all flow arrives at the sink within time T. Thus, it is required to model time horizon T in the solution techniques of MCC problem to be allowed to transfer the results of this problem to maximum dynamic flow problem. For this, an additional arc (d, s) from the sink d to the source s with sufficient capacity and transit time $-(T+1)$, is inserted on original network N.

We assign infinite capacity to the artificial arc in the transformed network N^+. However, flow along it is regulated by its adjacent real arc with capacity equal to that of the arc before it was split. Also, being zero transit time on this arc, the optimal MDCF computed on N^+ is not different with the optimal MDCF on original network N. The MDCF computation idea of [11] is applicable on N^+ that gives optimal solution and runs in strongly polynomial time. The order of time complexity of the algorithm is not affected by the increment of the size of the transformed network. Moreover, network transformation process in step 2 can be accomplished in linear time. Thus, for network $N = (V, E, c(e), \tau(e), s, d, T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs, Algorithm 1 solves discrete time maximum dynamic contraflow problem optimally in polynomial time, if the direction of the arcs are allowed to reverse only once at time zero.

Applying Algorithm 1 for the evacuation network N depicted in Figure 1. for time horizon $T = 10$, the maximum dynamic contraflow with discrete time setting is of value 52, whereas the maximum dynamic flow without contraflow is of value 30.

3.2. Maximum Dynamic Contraflow with Continuous Time Setting

Notion of natural transformation is helpful to generalize a discrete dynamic flow on a network $N = (V, E, c(e), \tau(e), s, d, T)$ as a continuous dynamic flow [15]. The notion states that the amount of flow that arrives at node w through arc $e = (v, w) \in E$ at time step θ in discrete time setting is equal to the amount of flow arriving at w through arc $e = (v, w)$ during unit interval of time $[\theta, \theta+1)$, i.e., $f_d(e, \theta) := f_c(e, [\theta, \theta+1))$ for all $\theta \in \{0, 1, \ldots, T-1\}$. Here, capacity constraints for continuous dynamic flow f_c are obviously obeyed, since $f_d(e, \theta) \leq c(e)$ implies $f_c(e, [\theta, \theta+1)) \leq c(e)$ for all time points in the interval $[\theta, \theta+1)$. This transformation is a bidirectional, if T and all transit time are integral [25]. We propose a solution procedure based on this notion for continuous time maximum dynamic contraflow (CT-MDCF) problem modeled on network N with not necessarily equal transit time on anti-parallel arcs and with integer inputs, if the direction of the arcs are allowed to reverse only once at time zero. The procedure has been summarized in Algorithm 2.

Algorithm 2: Algorithm CT-MDCF

1. Given network $N = (V, E, c(e), \tau(e), s, d, T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs.
2. Transform network N into N^+ as in Algorithm 1.
3. Transform network N^+ into its auxiliary network $\widetilde{N^+}$ as in [11].
4. Compute the discrete dynamic, temporally repeated flow on network $\widetilde{N}$ for time horizon $T-1$.
5. Transform the discrete dynamic flow into continuous dynamic flow using the natural transformation $f_d(e, \theta) := f_c(e, [\theta, \theta+1))$ for all $\theta \in \{0, 1, \cdots, T-1\}$.
6. Perform the flow decomposition into chain and cycle flows of the maximum flow obtained from step 4 and remove all cycle flows.
7. Arc $\overleftarrow{e} \in E$ is reversed if and only if the flow along arc $\vec{e} \in E$ is greater than $c(e)$ or if there is non-negative flow along arc $e \notin E$.
8. Get continuous time maximum dynamic contraflow on N for time horizon T.

Only the step 5 is additional effort in Algorithm 2 while comparing it with Algorithm 1. Since the transformation of $(T-1)$-horizon discrete time maximum dynamic flow yields a T-horizon continuous time maximum dynamic flow [15], the time complexity of finding a temporally repeated continuous flow is same to that in discrete case. Thus, for network $N = (V, E, c(e), \tau(e), s, d, T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs, Algorithm 2 solves continuous time maximum dynamic contraflow problem optimally in polynomial time, if the direction of the arcs are allowed to reverse only once at time zero.

Applying Algorithm 2 for the evacuation network N given in Figure 1. for time horizon $T = 10$, the maximum dynamic contraflow with continuous time setting is of value 45, whereas the maximum dynamic flow without contraflow is of value 26.

3.3. Earliest Arrival Contraflow on TTSP Network

Let $\mathrm{N}=(\mathrm{V}, \mathrm{E}, \mathrm{c}(\mathrm{e}), \tau(\mathrm{e}), s, d, T)$ be a network with not necessarily equal transit time on anti-parallel arcs and with integer inputs. The discrete time earliest arrival contraflow (DT-EACF) problem, if the direction of the arcs are allowed to reverse, maximizes the feasible net flow from s to d at each time steps $\theta \in \{0,1,\dots,\mathrm{T}\}$. In the following we propose an efficient solution procedure for DT-EACF problem modeled on a special class of network known as two terminal series-parallel (TTSP) network N. A TTSP network N is a network with a single source s and a single sink d which has a single arc (s,d) together with source s and sink d or is obtained from two series parallel networks N_1 and N_2 by one of the following two operations:

(i) Parallel Composition: Merge source nodes s_1 of N_1 and s_2 of N_2 to form the source s of N and merge sink nodes d_1 of N_1 and d_2 of N_2 to form the sink d of N.

(ii) Series Composition: Merge the sink node d_1 of N_1 with the source node s_2 of N_2.

We apply minimum cost flow algorithm of [26] to solve the DT-EACF problem for TTSP network. The algorithm solves maximum dynamic flow problem using a temporally repeated flow over the time horizon T. In fact, this maximum dynamic flow has the earliest arrival property [26]. We claim that two terminal series parallel network N, after transforming into network N^+, remains two terminal series parallel network. That is, the following algorithm solves the earliest arrival contraflow problem with discrete time setting, if the direction of the arcs are allowed to reverse only once at time zero. Moreover, the time complexity of Algorithm 3 is dominated by the polynomial time complexity of algorithm in [26].

Algorithm 3: Algorithm DT-EACF

1. Given TTSP network $\mathrm{N}=(\mathrm{V}, \mathrm{E}, \mathrm{c}(\mathrm{e}), \tau(\mathrm{e}), s, d, T)$ with not necessarily equal transit time on anti-parallel arcs and with integer inputs.
2. Transform network N into N^+ as in Algorithm 1.
3. Transform network N^+ into its auxiliary network $\widetilde{N^+}$ as in [11].
4. Solve earliest arrival flow problem on $\widetilde{N^+}$ by using the algorithm in [26].
5. Arc $\bar{e}$ is reversed if and only if the flow along arc $\vec{e}$ is greater than $\mathrm{c}(\mathrm{e})$ or if there is non-negative flow along arc $e \notin E$.
6. Get discrete time earliest arrival contraflow on N for time horizon T.

Together with the notion of natural transformation [15], Algorithm 3 solves continuous time earliest arrival contraflow problem on N when the arc reversal capability is allowed only once at time zero. Also, the solution to the problem is optimal and can be found in strongly polynomial time.

4. Conclusion

The importance and applicability of the idea of network contraflow especially in evacuation planning problem has been increasing due to its lane direction reversal capability. In the case of unequal to-and-fro transit time of oppositely directed lanes of a road, it is crucial to take network contraflow models with not necessarily equal transit time on anti-parallel arcs into account for preparing evacuation tasks. In this paper, we gave a network flow based evacuation model, an optimization model, capturing this situation. We studied the maximum dynamic contraflow problem and earliest arrival contraflow problem, and proposed strongly polynomial time algorithms as solution procedures, if the arcs are flipped only once at time zero, for discrete as well as continuous time setting. We also discussed about optimality and efficiency of the proposed algorithms, and present numerical examples that compares the optimal flow value with and without contraflow approach.

Applying the proposed model for real data-set and examine the performance of solution technique would enhance the scope of this work. Studying contraflow evacuation planning problems with other variants such as abstract contraflow, partial contraflow, lexicographically dynamic contraflows, etc., within the proposed model framework, are also crucial. Moreover, consideration of contraflow evacuation planning problems addressing the situation where multiple parallel lanes with different transit time exist on road topology would be further research work.

Acknowledgements

First author would like to thank University Grants Commission, Nepal for providing partial financial support (PhD Fellowship Award 2016) to conduct this research. Authors are grateful to anonymous reviewers whose comments significantly improved the quality of paper.

References

[1] Wei, Leyu, Jinliang Xu, Tian Lei, Menghui Li, Xingliang Liu, and Haoru Li. "Simulation and experimental analyses of microscopic traffic characteristics under a contraflow strategy." *Applied Sciences*. Vol. 9, No. 13, 2019, pp. 2651.

[2] Urbina, Elba, and Brian Wolshon. "National review of hurricane evacuation plans and policies: a comparison and contrast of state practices." *Transportation research part A: policy and practice*. Vol. 37, No. 3, 2003, pp. 257-275.

[3] Kim, Sangho, Shashi Shekhar, and Manki Min. "Contraflow transportation network reconfiguration for evacuation route planning." *IEEE Transactions on Knowledge and Data Engineering*. Vol. 20, No. 8, 2008, pp. 1115-1129.

[4] Ford Jr, Lester R., and Delbert Ray Fulkerson. "Constructing maximal dynamic flows from static flows." *Operations research*. Vol. 6, No. 3, 1958, pp. 419-433.

[5] Borradaile, Glencora, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. "Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time." *SIAM Journal on Computing*. Vol. 46, No. 4, 2017, pp. 1280–1303.

[6] Borrmann, André, Angelika Kneidl, Gerta Köster, Stefan Ruzika, and Markus Thiemann. "Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models." *Safety Science*. Vol. 50, No. 8, 2012, pp. 1695-1703.

[7] Göttlich, Simone, Sebastian Kühn, Jan Peter Ohst, Stefan Ruzika, and Markus Thiemann. "Evacuation dynamics influenced by spreading hazardous material." *Networks & Heterogeneous Media*. Vol. 6, No. 3, 2011, pp. 443-464.

[8] Khadka, Shree Ram and Phanindra Prasad Bhandari. "Model and solution for non-conservation flow evacuation planning problem." *The Nepali Mathematical Sciences Report*. Vol. 36, No. 1-2, 2019, pp. 11-16.

[9] Bhandari, Phanindra Prasad and Shree Ram Khadka. "Evacuation planning problems with intermediate storage." In *Proceedings of International Conference on Applied Mathematics and Computational Sciences (17-19 October, 2019)*, Dehradun, India. (To appear online).

[10] Bhandari, Phanindra Prasad, Shree Ram Khadka, Stefan Ruzika and Luca E Schäfer. "Lexicographically maximum dynamic flow with vertex capacities." *Journal of Mathematics and Statistics*. Vol. 16, No. 1, 2020, pp. 142–147.

[11] Rebennack, Steffen, Ashwin Arulselvan, Lily Elefteriadou, and Panos M. Pardalos. "Complexity analysis for maximum flow problems with arc reversals." *Journal of Combinatorial Optimization*. Vol. 19, No. 2, 2010, pp. 200-216.

[12] Anderson, Edward J., P. Nash, and Andrew B. Philpott. "A class of continuous network flow problems." *Mathematics of Operations Research*. Vol. 7, No. 4, 1982, pp. 501-514.

[13] Philpott, Andrew B. "Continuous-time flows in networks." *Mathematics of Operations Research*. Vol. 15, No. 4, 1990, pp. 640-661.

[14] Anderson, Edward J., and Andrew B. Philpott. "Optimisation of flows in networks over time." *Probability, Statistics and Optimisation*, F. P. Kelly, ed., Wiley, New York, ch. 27, 1994, pp. 369-382.

[15] Fleischer, Lisa, and Éva Tardos. "Efficient continuous-time dynamic network flow algorithms." *Operations Research Letters*. Vol. 23, No. 3-5, 1998, pp. 71-80.

[16] Koch, Ronald, Ebrahim Nasrabadi, and Martin Skutella. "Continuous and discrete flows over time." *Mathematical Methods of Operations Research*. Vol. 73, No. 3, 2011, pp. 301.

[17] Hashemi, S. Mehdi, and Ebrahim Nasrabadi. "On solving continuous-time dynamic network flows." *Journal of Global Optimization*. Vol. 53, No. 3, 2012, pp. 497-524.

[18] Khadka, Shree Ram, and Phanindra Prasad Bhandari. "Dynamic network contraflow evacuation planning problem with continuous time approach." *International Journal of Operations Research*. Vol. 14, No. 1, 2017, pp. 27-34.

[19] Pyakurel, Urmila, and Tanka Nath Dhamala. "Continuous dynamic contraflow approach for evacuation planning." *Annals of Operations Research*. Vol. 253, No. 1, 2017, pp. 573-598.

[20] Gale, David. "Transient flows in networks." *The Michigan Mathematical Journal*. Vol. 6, No. 1, 1959, pp. 59-63.

[21] Minieka, Edward. "Maximal, lexicographic, and dynamic network flows." *Operations Research*. Vol. 21, No. 2, 1973, pp. 517-527.

[22] Wilkinson, William L. "An algorithm for universal maximal dynamic flows in a network." *Operations Research*. Vol. 19, No. 7, 1971, pp. 1602-1612.

[23] Hoppe, Bruce. "Efficient dynamic network flow algorithms." PhD diss., Cornell University, 1995.

[24] Hoppe, Bruce, and Éva Tardos. "Polynomial time algorithms for some evacuation problems." In *SODA*. Vol. 94, 1994, pp. 433-441.

[25] Baumann, Nadine. "Evacuation by earliest arrival flows." PhD diss., University of Dortmund, Germany, 2007.

[26] Ruzika, Stefan, Heike Sperber, and Mechthild Steiner. "Earliest arrival flows on series-parallel graphs." *Networks*. Vol. 57, No. 2, 2011, pp. 169-173.

[27] Dhamala, Tanka Nath, and Urmila Pyakurel. "Earliest arrival contraflow problem on series-parallel graphs." International *Journal of Operations Research*. Vol. 10, No. 1, 2013, pp. 1-13.

[28] Dhungana, Ram Chandra and Tanka Nath Dhamala. "Maximum FlowLoc problems with network reconfiguration, *International Journal of Operations Research*. Vol. 16, No. 1, 2019, pp. 13–26.

[29] Dhungana, Ram Chandra, Urmila Pyakurel and Tanka Nath Dhamala. "Abstract contraflow models and solution procedures for evacuation planning." *Journal of Mathematics Research*. Vol. 10, No. 4, 2018, pp. 89–100.

[30] Pyakurel, Urmila, Hari Nandan Nath and Tanka Nath Dhamala. "Partial contraflow with path reversals for evacuation planning." *Annals of Operations Research*. Vol. 283, No. 1-2, 2019, pp. 591–612.

International Annals of Science
ISSN: 2456-7132
Volume 10, Issue 1, pp. 25-32, 2021
DOI: https://doi.org/10.21467/ias.10.1.25-32

RESEARCH ARTICLE

Maximum Flow Evacuation Planning Problem with Non-Conservation Flow Constraint

Phanindra Prasad Bhandari[1*], Shree Ram Khadka[2]

[1,2] Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal
[1] Department of Science and Humanities, Khwopa Engineering College, Bhaktapur, Nepal

* Corresponding author email: phanindra.maths@gmail.com

Received: 13 June 2020 / Revised: 02 August 2020 / Accepted: 20 August 2020 / Published: 04 September 2020

ABSTRACT

The optimization model of the maximum flow evacuation planning problem efficiently sends a maximum number of evacuees along with the routes of their transshipment from the disastrous zone, the source, to the safe zone, the sink, over a given time horizon. The limitation of the problem with the flow conservation constraint at the intermediate nodes is that even one more evacuee cannot be sent out from the source, if the evacuee cannot reach the sink. However, evacuators must attempt to send out as many evacuees as possible to safer places despite the sink. There may be relatively safe places in between the source and the sink. The limitation is due to the flow conservation constraint. In this paper, we remodel the problem with non-conservation flow constraint and propose an efficient algorithm. With this approach one can send as many evacuees as in the flow conservation case from the source to the sink. Moreover, a maximum number of evacuees can also be sent to the relatively safe places in between the source and the sink. The routes of their transshipment can also be identified.

Keywords: Evacuation Planning Problem, Pre-flow-push Algorithm, Network Flow, Disaster Management.

1 Introduction

Optimal use of the road network is required to implement efficient evacuation planning over a region in order to send a maximum number of evacuees from the disastrous zone, the source, to the safe zone, the sink, over a given time horizon [1]. The maximum flow evacuation planning problem efficiently sends a maximum number of evacuees from the source to the sink via various intermediate road segments through the road crossings as soon as possible during the evacuation or mitigation of the traffic in a rush hour in the crowded urban area. The motivation of researchers has been increased to study the problem so that the loss of people and their property could be minimized and that the traffic could be efficiently mitigated even in the crowded urban area during the rush hour.

The maximum flow problem on a single-source-single-sink static network, the network in which only the constant arc capacity is issued in each arc, has been studied in [2] and that on dynamic network, the network in which the constant arc capacity and transit time, i.e., the time a unit of flow takes to reach the neighboring node, has been studied in [3, 4]. A pseudo-polynomial algorithm based on the time-expanded network and a polynomial one based on the temporarily repeated flow with transit time on the arc as a cost coefficient have been investigated to solve the dynamic case of the problem. The solution procedure of [2] has been improved in [5] by choosing the shortest paths from the source to the sink in the residual network and in [6] augmenting all the shortest paths at once in each iteration in a layered sub-network of the residual network. Authors in [7] came up with an efficient procedure, known as push-relabel approach, for solving maximum flow problem that works on a node and its adjacent arcs at a time. From the 1950s to 2013, when Orlin [8] proved any instance of the problem can be solved in O(nm) time, various algorithms have been devised for solving this problem. A brief survey of the most important algorithms and their running time bounds is provided in [9]. The other variants of the evacuation planning problem have also been widely studied, [10--12].

The quickest flow problem, closely related to maximum dynamic flow problem, which minimizes the total time to send a given number of evacuees from a single source through a network to a single sink has been investigated in [13]. The algorithm can be executed in a polynomial time. Hoppe [14] extended the problem on a network with multiple sources and sinks. Another variant of the evacuation planning problem which sends a maximum evacuee at every step of time over time horizon is called the earliest arrival flow problem [15]. There exist exact algorithms with exponential time [16, 17]. A fully polynomial time approximation for the problem with a single-source-single-sink case has been investigated in [18], with multiple-sources-single-sink in [19--21] and that on series-parallel network in [22, 23]. A polynomial time algorithm for the problem with time dependent transit times and capacities is proposed in [24].

The evacuation planning problem with contraflow (arc reversal on network) approach has also been studied with a number of efficient solution procedures. The approach increases the outbound capacity of the arc and decreases the evacuation time. The maximum flow on a general static network, the maximum flow and the quickest flow cases on the dynamic network with single-source-single-sink have been investigated in [25] and the earliest arrival flow on a single-source-single-sink series-parallel network in [26] with efficient algorithms. The generalized maximum dynamic contraflow and generalized earliest arrival contraflow problems have been introduced and proposed pseudo-polynomial time algorithm for a single-source-single-sink lossy-network in [27]. The static lexicographic contraflow problem and the dynamic lexicographic contraflow problem with the constant travel time and node capacity that optimizes the feasible flow leaving or entering the terminals in the given order with polynomial time solutions have been investigated in [28].

The evacuation planning problem in continuous time setting has been introduced in [29] and extensively studied in [30--34] with a number of efficient algorithms. A strongly polynomial time algorithm to the maximum dynamic flow problem with contraflow approach in continuous time setting has been investigated in [35] with the arc reversal at time zero. Authors [36] discussed a number of dynamic contraflow problems in continuous time model with efficient algorithms. For deeper insight about evacuation planning problems, we refer to the survey articles [37, 38].

The existing optimization evacuation planning models with flow conservation constraint at the intermediate nodes has a limitation that even one more evacuee cannot be sent out from the source, if the evacuee cannot reach the sink. However, evacuators must attempt to send out as many evacuees as possible to safer places (despite the sink) in between the source and the sink where the evacuators can provide medical aids or other necessary supports to the evacuees or put the priorities among the evacuees to send to the sink. In this paper, we consider the maximum flow evacuation planning problem incorporating this aspect. We propose an efficient polynomial time solution procedure to solve the problem for static network and extend the solution procedure with a pseudo-polynomial time algorithm for single-source-single-sink dynamic network. Model of the problem and preliminary results without analytical proofs have been given in [39] also.

The paper is organized as follows. Mathematical formulation of the problem is described in Section 2. Section 3 contains solution to the problem with solution over static network in Subsection 3.1, on dynamic network in Subsection 3.2 and with priority based intermediate nodes in Subsection 3.3. An example is presented to implement the solution idea developed in this paper in Subsection 3.4. The last section concludes the paper.

Phanindra Prasad Bhandari *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 25-32, 2021

2 Problem Formulation

Consider a single-source-single-sink dynamic network $N = (V, E)$ where V and E are set of nodes and arcs, respectively, with $|V| = n$, $|E| = m$; $n - 1 \leq m$. The objective function of the maximum evacuation planning problem on N is to maximize

$$\boldsymbol{f} = \sum_{e \in \delta^-(d)} \sum_{\theta=0}^{T} f(e, \theta),$$

the total flow units of evacuees, over the given time horizon T, sent from the disastrous zone s, the source, to the safe zone d, the sink, through the possible routes reachable to the sink. The set of route segments e, the arc, entering into the sink d is denoted by $\delta^-(d)$. The integer flow unit $f(e, \theta)$, that flows at time step $\theta \in \{0, 1, \dots, T\}$ along the arc $e = (v, w)$ between the two adjacent intersections, the nodes, v to w such that $v, w \in V$, the set of intersections, follows the following constraints.

The flow units cannot exceed the arc capacity c_e for any time step within time horizon T, i.e.,

$$0 \leq f(e, \theta) \leq c_e \quad \forall\, e \in E \text{ and } \forall\, \theta \in \{0, 1, \dots, T\}. \qquad (1)$$

The flow units that enter into a node v for each time step may not exit from it at the same time or later within time horizon T, i.e.,

$$\sum_{e \in \delta^-(v)} \sum_{\theta=0}^{T-\tau_e} f(e, \theta) - \sum_{e \in \delta^+(v)} \sum_{\theta=0}^{T} f(e, \theta) \geq 0, \quad \forall\, v \in V \backslash \{s, d\}. \ (2)$$

where $\delta^-(v)$ and $\delta^+(v)$ denote for the set of arcs entering into the node v and leaving from it, respectively, and τ_e is the travel time for a flow unit along the arc $e \in E$.

The flow units remained at node v at time $\theta \in \{0, 1, \dots, T\}$, say the excess flow units $e_v(\theta)$, satisfies

$$e_v(\theta) \leq \sum_{e \in \delta^-(v)} f(e, \theta)\ \forall\, \theta \in \{0, 1, \dots, T\}. \ (3)$$

Such a node $v \in V \backslash \{s, d\}$ at which $e_v(\theta) > 0$ is said to be an active node at time θ and the corresponding flow $f(e, \theta)$ is a pre-flow.

3 Solution Procedure

3.1 Solution to the problem described on static network

Let us consider the problem described on the network $N = (V, E, c_e, s, d)$ with no transit time on its arcs. The solution procedure of the problem is based on the pre-flow-push algorithm investigated by Goldberg and Tarjan [7] with necessary modification. The algorithm works on a residual network $N_f = (V, E_f)$ of the network N with respect to the pre-flow f. If an arc $(v, w) \in E$ and $f(v, w) < c_{(v,w)}$, then E_f contains the arc (v, w) with residual capacity $r_{(v,w)} = c_{(v,w)} - f(v, w)$ and, if $(v, w) \in E$ and $f(v, w) > 0$, then E_f contains the arc (w, v) with residual capacity $r_{(w,v)} = f(v, w)$. The pre-flow $f(e)$ for the former and the latter case are said to be forward pre-flow and backward pre-flow, respectively.

A label function $l: V \rightarrow Z^+ \cup \{0\}$ is defined as

$$l(v) := \begin{cases} = |V| \ if\ v = s, \\ l(w) + 1\ if\ (v, w) \in N_f, \\ = 0\ if\ v = d. \end{cases}$$

If $l(v) = l(w) + 1$ with $e = (v, w) \in N_f$, $min\{e_v, c_e\}$ units of pre-flow are pushed from v to w. The node v with $e_v > 0$ is relabeled to be $l(v) := 1 + min\{l(w): (v, w) \in N_f\}$ when $l(v) \leq l(w)$ such that $l(v) \leq l(s)$. The initial label to the node $v \in V$ is assigned to be the shortest path distance of it from the sink d with the assumption of unit distance in each arc.

The pre-flow-push algorithm works on a node and its neighbors at a time and repeatedly selects an active node and pushes the flow to neighbors which are closer to the sink. The algorithm described in [7] sends the excess, the flow sent out from the source but is unable to reach to the sink, back to the source. If $l(v)$ is at least as large as $|V|$, the algorithm reaches a stage at which the excess flow at v must be pushed back towards the source. However, this seems to be not fair during evacuation rather to keep such flow units of evacuees at a temporary shelter, say v', at the intermediate node $v \in V$. This modifies the existing algorithm and becomes applicable to the evacuation planning problem.

This approach can be advantageous to provide immediate medication to severely injured evacuees, to keep the evacuees who lost their lives on the way at the temporary shelters and to make priorities among the evacuees to send to the sink. The procedure follows the operations as given in Algorithm 1.

Algorithm 1: Subroutine *PUSH-RELABEL-HOLD*(v)

Push: For an arc $(v, w) \in N_f$ of active node v with $l(v) = l(w) + 1$, push $\delta = \min\{e_v, c_{(v,w)}\}$ flow units along (v, w). Otherwise,
Relabel: For an active node v set $l(v) := 1 + \min\{l(w): (v, w) \in N_f\}$ satisfying $l(v) \leq l(s)$. Otherwise,
Hold: Hold excess $e_v > 0$ at v'.

Let us consider a static network $N = (V, E, c_e, s, d)$. The solution procedure (cf. Algorithm 2) to compute a maximum flow from the source node s to the sink node d is as follows.

Algorithm 2: *Modified Pre-flow-Push Algorithm for Static Network*

Input: Network $N = (V, E, c_e, s, d)$
Label Initialization: For all $v \in V \backslash \{s\}$, set $l(v)$ to be the shortest path distance of v from d and set $l(s) := n$.
Pre-flow Initialization: Set $f(e) := c_e$ for all $e \in \delta^+(s)$ and $f(e) := 0$ for remaining arcs.
Subroutine Application: Apply subroutine PUSH-RELABEL-HOLD (v) for each active node $v \in V$.
Output: Maximum static flow $\boldsymbol{f}$ with intermediate hold on N.

The maximum flow is decomposed into $s - d$ chains and that the excess flow, which is held at intermediate node v, is decomposed into $s - v$ chains for each $v \in V$. One could notice that no cycle of the flow is produced in each iteration on the residual network. The algorithm 2 yields optimal solution in strongly polynomial time.

Lemma 1. *Let $v \in V$ be an active node and $s \in V$ be the source node. There exists no pre-flow which could be pushed to s.*

Proof: Let the label at a node $v \in V$ be $l(v)$. A pre-flow f can be pushed from an active node v to its neighboring node w with $v, w \in V$ only if $l(v) = 1 + l(w)$. The Modified Pre-flow-Push algorithm always maintains the label $l(d) < l(s)$ at the sink d and $l(v) \leq l(s)$ at any intermediate node v.

Lemma 2. *Let $v \in V$ be an active node. One of the operations, push or relabel or hold, is applicable to the node v.*

Proof: Let l and f be any valid labeling and a pre-flow at a node in the residual network N_f, respectively. For an active node $v \in V$, there exists an arc $(v, w) \in N_f$ such that $l(v) \leq l(w) + 1$. The node v is relabeled if $l(v) < l(w) + 1$ and a pre-flow is pushed to the node w if $l(v) = l(w) + 1$. If the pre-flow cannot be pushed elsewhere then it is held at the temporary shelter v' of the node v.

Theorem 1. *The Modified Pre-flow-Push Algorithm yields a maximum flow on network N.*

Proof: At an active node $v \in V$, either there is a push of a pre-flow to a neighboring node or the node is relabeled or the excess is held at the temporary shelter v'. The pre-flow from s to d is feasible. If a pre-flow can neither be pushed nor the node be relabeled, the excess is held at the shelter v' of the node v. This makes the node v to be inactive. The algorithm is executed when there remains no active node in the residual network and pre-flow becomes a flow. The algorithm is terminated when there does not exist any $s - d$ chain in N_f. Moreover, a flow is maximum if and only if there is no chain from s to d in N_f, [2].

Theorem 2. *The subroutine PUSH-RELABEL-HOLD runs in strongly polynomial time.*

Proof: Consider a network N with n nodes and m arcs such that $n - 1 \leq m$. The number of relabeling operations is at most n for each $n - 2$ nodes. That is, relabeling can be done in $O(n^2)$ times overall. Number of saturating pushes of pre-flow is at most n per arc for at most m arcs and number of non-saturating pushes is at most $n^2m + n(n - 2)$ over the network, [7]. Thus, the push operations can be accomplished in $O(n^2m)$ times, a dominating bound of the subroutine. The hold operations do not exceed the relabeling iterations.

Phanindra Prasad Bhandari *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 25-32, 2021

Furthermore, the push operation can be improved with time $O(n^2m)$ by using the first-in, first-out algorithm or the maximum-distance method described in [7]. The dynamic tree data structure modifies the algorithm with time $O(nm\, log(n^2/m))$, [40].

Computation of distance in Algorithm 2 can be accomplished in time $O(m+n)$ by using the breadth first search method on the inverse network. The maximum and the excess flow can be decomposed into chains in $O(mn)$ time, [41]. This is an additional effort to implement the algorithm for the maximum flow evacuation planning problem. Hence one can observe that the algorithm is strongly polynomial.

3.2 Solution to the problem described on dynamic network

The solution procedure for the static network can also be applied to the dynamic case of the problem described on the dynamic network $N = (V, E, c_e, \tau_e, s, d, T)$. The time-expanded network N^T, suggested in [3], of network N for time horizon T without holdover arcs is given by $N^T = (V^T, E^T, c_e, s, d)$ where $V^T = \{v(\theta): v \in V, \theta \in \{0, 1, \dots, T\}\}$ and $E^T = \Big\{\Big(v(\theta), w\big(\theta + \tau_{(v,w)}\big)\Big): v \neq w, v, w \in V \wedge \theta \in \big\{0, 1, \dots, T - \tau_{(v,w)}\big\}\Big\}$. For the modification, the sets $\{(s(\theta), s(\theta+1);\ \theta \in \{0, 1, \dots, T-1\}\}$ and $\{(d(\theta), d(\theta+1); \theta \in \{0, 1, \dots, T-1\}\}$ of arcs with sufficient capacities are added to the set E^T. Moreover, each node $v(T); v \in V \backslash \{s\}$ is connected to $d(T)$ by an artificial arc with zero capacity. The exact solution procedure that solves the maximum dynamic flow problem is given in Algorithm 3.

Algorithm 3: ***Modified Pre-flow-Push Algorithm for Dynamic Network***

1. Given network $\mathrm{N} = (\mathrm{V}, \mathrm{E}, \mathrm{c_e}, \tau_\mathrm{e}, \mathrm{s}, \mathrm{d}, \mathrm{T})$.
2. Find time expanded network $\mathrm{N}^\mathrm{T}$of N.
3. Apply "Modified Pre-flow-Push Algorithm for Static Network" on $\mathrm{N}^\mathrm{T}$where $\mathrm{s}(0)$ is the source and $\mathrm{d}(\mathrm{T})$ is the sink.
4. Get maximum dynamic flow with intermediate hold on N.

The solution on the dynamic network can be obtained in a polynomial time with respect to the input size. In particular, the relabel operation, the push operation and the hold operation can be executed in (T^2n^2), $O(T^3n^2m)$ and $O(T^2n^2)$ times, respectively which are pseudo-polynomial time complexities.

3.3 Solution to the problem with priority based intermediate nodes

The intermediate nodes, in which the excess flow of evacuees might be held, may not be of equal importance with respect to the risk, the distance from the sink, the holding capacity, etc. The intermediate nodes on the residual network N_f after computing the maximum static flow $\boldsymbol{f}$ be ordered as $I_1, I_2, \dots, I_{(n-2)}$ from lower priority to higher one.

We consider a residual sub-network N_1obtained from N_f by deleting the source node s, the sink node d and the arcs leaving from s and that entering into d. The previous labels on the nodes of N_1 are removed and the excess flow at I_1 is pushed to $I_i\,; i = (n-2), (n-3), \dots, 2,$ using Algorithm 2 after slight modification. For, we push the flow from the new source (i.e., I_1) that equals to the minimum of excess at I_1 and the sum of the capacities of the arcs leaving it. The procedure is repeated on N_2, the residual sub-network obtained from $N_1$by deleting the node I_1 and the arcs leaving from it and so on until the maximum flow from $I_{(n-3)}$ to $I_{(n-2)}$ is procured. The algorithm is executed $(n-3), (n-4), \dots, 2, 1$ times on the residual sub-networks $N_1, N_2, \dots, N_{(n-3)}$, respectively for priority-based holding purpose. Hence the procedure works within $O(n^4m)$ time.

3.4 Implementation of the algorithm

Let us consider the network depicted in Figure 1 for a region with 28 intersections as nodes and 83 road segments as arcs with the source node S and the sink node D and with specified road capacity and the transit time, for example, the arc capacity and the transit time for the arc from the node S to the node a are 4 and 1, respectively. The zero-transit time means the two nodes are close enough with negligible transit time. Not all routes from the source to the sink out of 174,007 possible routes are equally efficient as the arc capacity and the transit time vary in each arc. The algorithm finds the efficient routes, maximum flow that reaches to the sink and maximum flow that reaches to the intermediate nodes with the specific nodes.

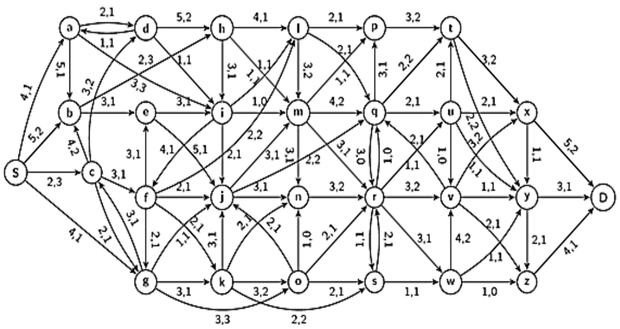


Figure 1: Evacuation Network.

Result

We have considered the static, the priority based static and the dynamic networks extracted from the Figure 1. 12 units of evacuees as the maximum flow reached into the sink D, 2 units at the node e and 1 unit at x on the static network. The evacuees placed in the intermediate nodes can be sent to the prioritized node, for example, the priority order is z, y, x, w with the assumption that the nodes near the sink are relatively safe. The flow placed at the node e can be rerouted to keep one unit of evacuee at the node y and one unit at the node z.

The time expanded network developed from the dynamic network in Figure 1, for time horizon $T = 10$, consists of 299 nodes and 820 arcs. 20 units of maximum flow and the optimal routes from the source to the intermediate nodes with 121 units of maximum flow to be placed at the end node of the routes.

It took around 2 seconds to run the program for example we considered for time horizon $T = 10$. It takes 201.50 seconds and 1226.27 seconds if we set $T = 30$ and $T = 50$, respectively. Detailed output for the latter cases has been omitted here. The algorithm has been coded into Python with version 3.6.1 and was run on the computer with 2.00 GB RAM and i3@2.30 GHZ of processor.

4 Conclusion

The evacuation planning problem with flow conservation at intermediate nodes does not allow to send evacuees from risk zone to any other intermediate spots though these are relatively safe even in the case of all evacuees not being able to reach the destination. Thus, the problem that allows to hold evacuees in such spots, despite sending a maximum flow to the safe destination, is important to investigate for the evacuation planning. In this paper, we proposed the maximum flow evacuation planning problem with relaxed flow conservation constraints at the intermediate nodes with its mathematical formulation and efficient solution procedure. This approach allows holding the evacuees in the relatively safe places during evacuation. The solution procedure is based on the pre-flow-push algorithm, one of the efficient maximum flow algorithms, with necessary modification. Moreover, the algorithm has been extended to work on the dynamic network using the modified time expanded network approach and on the network in which the intermediate nodes are in priority order. The proposed procedure is simple to implement as we described in Subsection 3.4. The limitation associated to current work is that the algorithm solving for dynamic version of the evacuation flow problem leads to a pseudo-polynomial time complexity. The maximum flow aspect of the problem defined for multiple-sources-multiple-sinks network and the other aspects, such as the earliest arrival flow, the quickest flow of the problem, are interesting area for further research.

5 Declarations

5.1 Acknowledgements

The research of first author is partially supported by University Grants Commission, Nepal under the PhD Fellowship Award scheme (Award No. PhD-72/73 S&T-01) 2016.

5.2 Competing Interests

The authors have no conflicts of interest regarding this paper.

How to Cite this Article:

Phanindra Prasad Bhandari and Shree Ram Khadka "Maximum Flow Evacuation Planning Problem with Non-Conservation Flow Constraint", *Int. Ann. Sci.*, vol. 10, no. 1, pp. 25-32, Sep. 2020.

References

[1] A. Kimms and K. C. Maassen, "Optimization and simulation of traffic flows in the case of evacuating urban areas", *OR Spectrum*, vol. 33, pp. 571-593, 2011.

[2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network", *Can. J. Math.*, 8, pp. 399-404, 1956.

[3] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows', *Operations Research*, vol. 6, pp. 419-433, 1958.

[4] L. R. Ford and D. R. Fulkerson, "*Flows in Networks*", Princeton University Press, 1962.

[5] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the Association for Computing Machinery*, vol. 19, pp. 248-264, 1972.

[6] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in a network with power estimation", *Soviet Math. Dokl*, vol. 11, pp. 1277-1280, 1970.

[7] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem", *Journal of the Association for Computing Machinery*, vol. 35, pp. 921-940, 1988.

[8] J. B. Orlin, "Max flows in O (nm) time, or better", In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 765-774, 2013.

[9] P. Kovacs, "*Efficient Algorithms for Graph Optimization Problems*", PhD thesis, Eotvos Lorand University, Hungary, 2019.

[10] H. W. Hamacher, S. Heller, W. Klein, G. Köster, and S. Ruzika, "A sandwich approach for evacuation time bounds", In *Pedestrian and Evacuation Dynamics*, pp. 503-513. Springer, Boston, MA, 2011.

[11] A. Borrmann, A. Kneidl, G. Köster, S. Ruzika, and M. Thiemann, "Bidirectional coupling of macroscopic and microscopic pedestrian evacuation models", *Safety science*, vol. 50, pp. 1695-1703, 2012.

[12] G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum and C. Wulff-Nilsen, "Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time", *SIAM Journal on Computing*, vol. 46, pp. 1280-1303, 2017.

[13] R. Burkard, K. Dlaska and B. Klinz, "The quickest flow problem", *ZOR-Methods and Models of Operations Research*, vol. 37, pp. 31-58, 1993.

[14] B. Hoppe, "*Efficient Dynamic Network Flow Algorithms*", PhD Thesis, Cornell University, 1995.

[15] D. Gale, "Transient flows in networks", *The Michigan Mathematical Journal*, vol. 6, pp. 59-63, 1959.

[16] E. Minieka, "Maximal, lexicographic, and dynamic network flows", *Operations Research*, vol. 21, pp. 517-527, 1973.

[17] W. L. Wilkinson, "An algorithm for universal maximal dynamic flows in a network", *Operations Research*, vol. 19, pp. 1602-1612, 1971.

[18] B. Hoppe and E. Tardos, "Polynomial time algorithms for some evacuation problems", In *Proccedings of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pp. 433-441, 1994.

[19] N. Baumann, "*Evacuation by Earliest Arrival Flows*", PhD Thesis, University of Dortmund, Germany, 2007.

[20] N. Baumann and M. Skutella, "Solving evacuation problems efficiently: earliest arrival flows with multiple sources", In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 399-410, 2006. IEEE.

[21] N. Baumann and M. Skutella, "Earliest arrival flows with multiple sources", *Mathematics of Operations Research*, vol. 34, pp. 499-512, 2009.

[22] M. Steiner, "*A Survey on Earliest Arrival Flows and a Study of the Series-parallel Case*", Diploma Thesis, University of Kaiserslautern, Germany, 2009.

[23] S, Ruzika, H. Sperber and M. Steiner, "Earliest arrival flows on series-parallel graphs", *Networks*, vol. 57, pp. 169-173, 2011.

[24] S. A. Tjandra, "*Dynamic Network Optimization with Application to the Evacuation Problem*", PhD Thesis, University of Kaiserslautern, Germany, 2003.

[25] S. Rebennack, A. Arulselvan, L. Elefteriadou and P. M. Pardalos, "Complexity analysis of maximum flow problem with arc reversal", *Journal of Combinatorial Optimization*, vol. 19, pp. 200-216, 2010.

[26] T. N. Dhamala and U. Pyakurel, "Earliest arrival contraflow problem on series-parallel graphs", *International Journal of Operations Research*, vol. 10, pp. 1-13, 2013.

[27] U. Pyakurel, H. W. Hamacher and T. N. Dhamala, "Generalized maximum dynamic contraflow on lossy network", *International Journal of Operations Research Nepal*, vol. 3, pp. 27-44, 2014.

[28] U. Pyakurel and T. N. Dhamala, "Models and algorithms on contraflow evacuation planning network problems", *International Journal of Operations Research*, vol. 12, pp. 36-46, 2015.

[29] A. B. Philpott, "*Algorithms for Continuous Network Flow Problems*", PhD Thesis, University of Cambridge, UK, 1982.

[30] E. J. Anderson, P. Nash and A. B. Philpott, "A Class of Continuous Network Flow Problems", *Mathematics of Operations Research*, vol. 7, pp. 501-514, 1982.

[31] A. B. Philpott, "Continuous-time flows in networks", *Mathematics of Operations Research*, vol. 15, pp. 640-661, 1990.

[32] R. Koch, E. Nasrabadi and M. Skutella, "Continuous and discrete flows over time: a general model based on measure theory", *Mathematical Methods of Operations Research*, vol. 73, pp. 301-337, 2011.

[33] S. M. Hashemi and E. Nasrabadi, "On solving continuous-time dynamic network flows", *Journal of Global Optimization*, vol. 53, pp. 497-524, 2012.

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

[35] S. R. Khadka and P. P. Bhandari, "Dynamic network contraflow evacuation planning problem with continuous time approach", *International Journal of Operations Research*, vol. 14, pp. 27-34, 2017.

[36] U. Pyakurel and T. N. Dhamala, "Continuous dynamic contraflow approach for evacuation planning", *Annals of Operations Research*, vol. 253, pp. 573-598, 2017.

[37] M. Skutella, "An introduction to network flows over time", In: *Research trends in combinatorial optimization*, pp. 451-482. Springer, Heidelberg 2009.

[38] T. N. Dhamala, "A survey on models and algorithms for discrete evacuation planning network problems", *Journal of Industrial and Management Optimization*, vol. 11, pp. 265-289, 2015.

[39] S. R. Khadka and P. P. Bhandari, "Model and solution for non-conservation flow evacuation planning problem", *The Nepali Mathematical Sciences Report*, vol. 36, pp. 11-16, 2019.

[40] D. D. Sleator and R. E. Tarjan, "A data structure for dynamic trees", *Journal of Computer and System Sciences*, vol. 26, pp. 362-391, 1983.

[41] R. K. Ahujia, T. L. Magnanti and J. B. Orlin, "*Network Flows: Theory, Algorithms and Applications*", New Jersey: Rentice-Hall, 1993.

ISSN 2091-1521 (Print)
2091-153X (Online)
International Journal of Operational Research/Nepal - IJORN - 2020, Vol. 9, Issue 1

Lexicographic Earliest Arrival Contraflow Evacuation Problem on UPL-TTSP Network

Shree Ram Khadka[1] and Phanindra Prasad Bhandari[2*]

[1,2]Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal
Email(s): [1]shree.khadka@cdmath.tu.edu.np; [2]phanindra.maths@gmail.com

Abstract

An evacuation plan proposes a way out for maximum utilization of an existing transportation system so that the safest tour and most efficient evacuation time of all expected evacuees could be ensured. This paper studies the auto-based earliest arrival contraflow problem modeled on network with capacitated intermediate vertices. The vertices serve as transshipment vertices as well as temporary shelters of given priority order. It also proposes an efficient solution procedure to the problem for the uniform path length two terminal series parallel (UPL-TTSP) network.

Keywords: Network contraflow, Capacitated vertices, Evacuation planning problem, Disaster management

1. Introduction

Increasing number of disasters due to different kind of natural and human caused hazards have drawn attention of stakeholders for preparation of optimal evacuation plans. Evacuation planning problems can be modeled as network optimization flow problems. Since it is usually not known when disaster will actually happen, it is desirable to organize an evacuation in such a way that as many evacuees as possible are sent to the sink as early as possible. Moreover, it is not always possible to send all the evacuees into the sink due to network capacity constraint and/or time constraint, but can be held at relatively safe intermediate spots besides sending to the sink. One of the potential ways to accelerate the process of achieving this objective is to apply notion of network contraflow in evacuation model with non-conservation flow constraints.

Maximum dynamic flow (MDF) problem, introduced by Ford and Fulkerson [11, 12] is a central problem in network flow theory and in mathematical evacuation planning problem that attempts to determine maximum flow of single commodity from a single source to a single sink over specified time horizon. The problem with arc direction reversal capability (contraflow), known as maximum dynamic contraflow (MDCF) problem, has been studied and proposed polynomial time solution algorithm in [18]. The MDCF problem with continuous time setting has been studied in [15]. Authors in [5] consider not necessarily equal transit time on anti-parallel arcs and study MDCF problem for general network for discrete as well as continuous time setting.

An extension of MDF problem is earliest arrival flow (EAF) problem. An EAF problem aims to optimize the objective of MDF problem for every time step within pre-specified time horizon. Existence of solution to earliest arrival flow problem was proved by Gale [13]. The problem has been studied in different contexts since then, see [2], [3], [14], [17], [20], [21], etc. To the best of our knowledge no polynomial time exact algorithm for earliest arrival flow problem on general network is known till now. Ruzika et al. [19] considered maximum dynamic flow problem on special class of networks known as two terminal series parallel (TTSP) network. They developed a polynomial time MDF algorithm for TTSP networks and showed that this also solves earliest arrival flow problem for these networks. Authors also pointed out that on series-parallel networks it is always possible to find an earliest arrival flow as a temporally repeated flow. However, the converse is not necessarily true. Earliest arrival contraflow (EACF) problem has been studied and proposed a polynomial time solution algorithm, based on the idea in [19], in [10] for TTSP network. Authors in [5] consider not necessarily equal transit time on anti-parallel arcs to study EACF problem for this class of networks. Network flow problem with non-conservation flow constraints is studied in the context of evacuation planning problem in [9] (cf. [4], [6], [16]). By considering the multinetwork the contraflow evacuation problems with capability of holding evacuees at intermediate spots have been studied in [8]. Authors have proposed polynomial time solution algorithm for static case and pseudo-polynomial time solution algorithm for dynamic case.

This paper studies the earliest arrival contraflow problem modeled on the network that allows holding of flow units at intermediate vertices. It discusses solution procedure for the problem modeled for uniform path length two terminal series parallel (UPL-TTSP) network. The problem is formulated in Section 2. Section 3 discusses its solution procedure. Section 4 concludes the paper.

2. Problem Formulation

Consider a network $N = (V, A, u(a), \tau(a), k(v), s, d)$ with vertex set V and arc set A, both to be finite, such that $n: = |V|$ and $m: = |A|$. Vertices s and d represent the source and the sink, respectively. Here, $u: A \rightarrow \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ represent the arc capacity function which bounds the number of flow units on each arc $a \in A$ at each time step $t \in \mathcal{T} := \{0, 1, \dots, T\}$ where T is given time horizon. Moreover, the transit time function $\tau: A \rightarrow \mathbb{N}$ specifies the time needed by a flow unit to traverse an arc. We assume a terminal set $S: = \{v_1, \dots, v_r\} \subset V$ with prioritized from higher to lower priority, i.e., $d = v_1 \succcurlyeq \cdots \succcurlyeq v_r$, to be given. Similarly, the vertex capacity function $k: S \rightarrow \{0, \infty\}$ delimits the total number of flow units, which may be held in each of the vertices $v_i \in S$.

The nonnegative variables $f(a,t)$ defined by $f: A \times \mathcal{T} \rightarrow \mathbb{N}_0$ is the number of flow units entering arc a at time step t that specify the flow over time in the network N. This flow unit should be bounded by the capacity of arc, i.e., $f(a,t)$ satisfies the capacity constraints for all $a \in A$ and for all $t \in \mathcal{T}$. That is,

$$0 \leq f(a,t) \leq u(a) \qquad \forall\, a \in A \text{ and } t \in \mathcal{T}. \tag{1}$$

Moreover, $f(a,t)$ has to be equal to zero for all $t > T - \tau(a)$ for all $a \in A$. The excess flow at vertex $v \in V$ at time $t \in \mathcal{T}$, denoted by $ex_f(v,t)$, is defined as

$$\begin{aligned} 0 \leq ex_f(v,t) := & \sum_{a \in \delta^{-(v)}} \sum_{\xi=0}^{t-\tau(a)} f(a,\xi) \\ & - \sum_{a \in \delta^{+(v)}} \sum_{\xi=0}^{t} f(a,\xi) \end{aligned} \tag{2}$$

where $\delta^{-(v)} := \{ a \in A : a = (w,v)$ for some vertex $w \in V\}$ and $\delta^{+(v)} := \{ a \in A : a = (v,w)$ for some vertex $w \in V\}$ denote the set of arcs entering and leaving vertex $v \in V$, respectively.

Further, we need to ensure that the excess flow at each vertex $v \in S$ over time horizon T is to be bounded by the capacity (v) , i.e.,

$$ex_f(v,T) \leq k(v) \text{ for all } v \in S. \tag{3}$$

Consequently, the total flow of evacuees leaving the source s equals the total flow of the evacuees held at vertices $v \in S$ over the time horizon T, i.e.,

$$\begin{aligned} \sum_{a \in \delta^{+(s)}} \sum_{\xi=0}^{T} f(a,\xi) - & \sum_{a \in \delta^{-(s)}} \sum_{\xi=0}^{T} f(a,\xi) \\ & = \sum_{v \in S} ex_f(v,T). \end{aligned} \tag{4}$$

With these settings, *lexicographic earliest arrival contraflow (LexEACF) problem* on N aims to maximize the flow units sent to the terminals on S in given priority order at each time step $t \in \mathcal{T}$, if reversal of arc direction is allowed. An arc $a = (v,w) \in A$ in which the flow could travel from vertex v to vertex w is replaced by the arc (w,v) for contraflow purpose.

3. Solution Discussion

The solution procedure for the lexicographic earliest arrival contraflow problem for a UPL-TTSP network N is discussed here. A directed dynamic network is a uniform path length (UPL) network for which the sum of the transit times on arcs on any possible path from the source s to the vertex v, for all $v \in V$, is equal. A two terminal series-parallel network $N = (V, A)$ is a directed network with a single source s and a single sink d which has a single arc (s, d) or is obtained from two series parallel networks $N_1$and N_2 by one of the two operations: Parallel Composition and Series Composition. The first suggests to merge source vertices s_1 of $N_1$and s_2 of N_2 to form the source vertex s of N and merge sink vertices d_1 of $N_1$and d_2 of N_2 to form the sink vertex d of N. The second suggests to merge the sink vertex d_1 of $N_1$with the source vertex s_2 of $N_2$to form the network N with source vertex s_1 and sink vertex d_2.

To solve the LexEACF problem, if arcs reversibility is allowed only once at time zero, the input network is modified to its corresponding auxiliary network by the technique given in [18] and a maximum dynamic flow is computed on it by using the technique given in [6]. For the modification of input network $N = (V, A, u(a), \tau(a), s, d, T)$, it is transformed into its auxiliary network $\widetilde{N} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), k(v), s, d, T)$ where arc set $\tilde{A}$ contains undirected arc (v, w) , if $(v, w) \in A$ and/or $(w, v) \in A$ such that capacity $u(\widetilde{v, w}) := u(v, w) + u(w, v)$ and transit time $\tau(\widetilde{v, w}) := \tau(v, w)$ if $(v, w) \in A$, otherwise, $\tau(\widetilde{v, w}) := \tau(w, v)$. During the dynamic flow computation, a lexicographic dynamic flow is computed on so-formed auxiliary network $\widetilde{N}$ by using the Lexicographically maximum dynamic flow (LexMDF) algorithm of [7]. The maximum dynamic $s - v_i$ flow on $\widetilde{N}$ is obtained iteratively by using temporally repeated flows (TRFs) on it. The TRF is a dynamic flow that can be generated by repeating all possible source to sink path flows starting at time zero and then adopting temporal repetition as far as possible.

For a uniform path length (UPL) network $N = (V, A, u(a), \tau(a), k(v), s, d, T)$, with prioritized vertices $v_i \in V$ sorted as $d = v_1 \succcurlyeq \cdots \succcurlyeq v_r$, the procedure for solving LexMDF problem has following steps: The main idea of the solution procedure of the problem is to find $s - v_i$ paths, for all $v_i \in V: k(v_i) > 0$, at all possible time steps $t \in \mathcal{T}$ with corresponding flow value and send as many units of flow as possible along the paths as long as possible. Such paths can be found by decomposing the flow on solving the *lexicographic minimum cost flow (LexMinCF) problem* on N, see [7]. It gives an extended set $\Gamma_{v_i}^E$ that contains all minimum cost $s - v_i$ paths that exist at any time $t \in \mathcal{T}$ on the residual network of N with respect to the optimal flow $f(v_{i-1})$ at previous immediate prioritized vertex v_{i-1}. Flow units of corresponding values are pushed along each path as long as possible. Moreover, the flow is pushed along the paths in $\Gamma_{v_i}^E$ with the strategy of saving unused paths for the use of next less prioritized vertex v_{i+1} without violating the optimality at v_i. This is assured by selecting the path with highest $F_t\ (\gamma_{v_i})$, the time step at which the flow along γ_{v_i}stops to get repeated, among the paths $\gamma_{v_i} \in \Gamma_{v_i}^E$ with highest $I_t\ (\gamma_{v_i})$, the time step at which the flow along γ_{v_i} starts to get repeated, at the first and so on. This procedure yields an optimal solution to the LexMDF problem on UPL network N in polynomial time.

Here the LexMDF algorithm is applied for UPL-TTSP network after reducing it into its auxiliary network that modifies the LexEAF algorithm proposed in [7] and solves the LexEACF problem. The modified procedure is given in Algorithm 1 that solves LexEACF problem when arc reversibility on N is allowed only once at time zero.

Algorithm 1. Lexicographic Earliest Arrival Contraflow Algorithm

1. Given a UPL-TTSP network $N = (V, A, u(a), \tau(a), k(v), s, d, T)$, $S := \{v_1, \dots, v_r\} \subset V$ with $d = v_1 \succcurlyeq \cdots \succcurlyeq v_r$ and integer inputs.

2. Transform N into $\widetilde{N} = (V, \tilde{A}, u(\tilde{a}), \tau(\tilde{a}), k(v), s, d, T)$ as in [18].

3. Compute LexMDF on $\widetilde{N}$ using Algorithm in [7]

4. Perform flow decomposition into path and cycle flows of maximum flow obtained from step-3 and remove all cycle flows.

5. Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $\notin A$.

6. Obtain LexEACF solution for UPL-TTSP network N.

We make the following claim which turns out to be important in proving the optimality of solution computed by Algorithm 1.

Claim 1. UPL-TTSP network N, after transforming into its auxiliary network $\widetilde{N}$, remains UPL-TTSP network.

Theorem 1. Given a UPL-TTSP network $N = (V, A, u(a), \tau(a), k(v), s, d, T)$, source s and terminal set $S := \{v_1, \dots, v_r\} \subset V$ with $d = v_1 \succcurlyeq \cdots \succcurlyeq v_r$. Then, Algorithm 1 computes a lexicographic earliest arrival contraflow on N in polynomial time.

Proof: The construction of auxiliary network $\widetilde{N}$ of input network N is well defined and the transformed network remains a UPL-TTSP network due to Claim 1. The algorithm (Step 3) pushes flow of value $f(\gamma_{v_i})$ along each path on extended set $\Gamma_{v_i}^{E}$ for each possible time step $t \in \{0, 1, 2, \dots, T - \tau(\gamma_{v_i})\}$ from the source vertex s to each of the destination vertex $v_i \in S$ in given priority order [6]. Therefore, a maximum flow at each $v_i \in S$ is obtained at the termination of algorithm, [12]. Moreover, the network N being a two terminal series parallel in structure, this flow has an earliest arrival property [19].

The time complexity of Algorithm 1 depends on time complexity of the solution procedure on the reduced network $\widetilde{N}$. Also, it is dominated by solving a LexMDF problem on $\widetilde{N}$ since the flow decomposition in each iteration and network transformation can be done only in $O(mn)$ ([1]) and $O(m)$ time, respectively. The LexMDF problem can be solved in strongly polynomial time, [7].

4. Conclusion

Earliest arrival flow problems are of much interest to evacuators because these problems ensure that the number of evacuated persons is maximum at each time point within given time horizon T. Evacuation models with intermediate temporary shelters could be extra benefit while implementing them and contraflow approach seems to be a crucial tool to speed up the overall evacuation process during disasters. This paper proposed a network contraflow evacuation model that allows holding of evacuees at intermediate spots despite sending them to the safe destination. It also proposed polynomial time solution algorithm to the lexicographic earliest arrival contraflow problem for UPL-TTSP network.

Searching of solutions to the problem at which the arc reversibility is permitted at any time point within the specified time horizon would be further research in the field considered here. Problem modeled for multinetwork would also be interesting that captures the situation with multiple lanes connecting two places with unequal transit time on them.

Acknowledgments: Second author would like to thank University Grants Commission, Nepal for providing partial financial support as a PhD Fellowship Award-2016 (Award No.: PhD-72/73-S\&T-01) to conduct this research.

References

[1] Ahuja RK, Magnati TL and Orlin JB (1993) *Network flows: theory, algorithms, and applications*, Prentice Hall, New Jersey.

[2] Baumann N (2007) *Evacuation by earliest arrival flows*, Doctoral thesis, Department of Mathematics, University of Dortmund, Germany.

[3] Baumann N and Skutella M (2009) Earliest arrival flows with multiple sources, *Mathematics of Operations Research* **34** 499--512.

[4] Bhandari PP and Khadka SR (2020) Evacuation planning problems with intermediate storage, In *AIJR Proceedings of International Conference on Applied Mathematics & Computational Sciences (ICAMCS-2019), Dehradun*, 90-95.

[5] Bhandari PP and Khadka SR (2020) Evacuation contraflow problems with not necessarily equal transit time on anti-parallel arcs, *American Journal of Applied Mathematics*} **8**(4) 230--235.

[6] Bhandari PP and Khadka SR (2020) Maximum flow evacuation planning problem with non-conservation flow constraint, *International Annals of Science*} **10**(1) 25--32.

[7] Bhandari PP and Khadka SR (2020) Evacuation planning problems on uniform path length network with prioritized destinations, (Submitted)

[8] Bhandari PP and Khadka SR (2020) Lexicographically maximum contraflow problems with vertex capacities, (Submitted)

[9] Bhandari PP, Khadka SR, Ruzika S and Schafer LE (2020) Lexicographically maximum dynamic flow with vertex capacities, *Journal of Mathematics and Statistics* **16**(1) 142--147.

[10] Dhamala TN and Pyakurel U (2013) Earliest arrival contraflow problem on series-parallel graphs, *International Journal of Operations Research* **10**(1) 1--13.

[11] Ford LR and Fulkerson DR (1958) Constructing maximal dynamic flows from static flows, *Operations research* **6**(3) 419--33.

[12] Ford LR and Fulkerson DR (1962) *Flows in networks*, Princeton University Press.

[13] Gale D (1959) Transient flows in networks, *Michigan Mathematical Journal* **6** 59-63.

[14] Hoppe B and Tardos E (1994) Polynomial time algorithms for some evacuation problems, In *SODA* **94** 433--441.

[15] Khadka SR and Bhandari PP (2017) Dynamic network contraflow evacuation planning problem with continuous time approach, *International Journal of Operations Research* **14**(1) 27-34.

[16] Khadka SR and Bhandari PP (2019) Model and solution for non-conservation flow evacuation planning problem, *The Nepali Mathematical Sciences Report*} **36** 11-16.

[17] Minieka E (1973) Maximal, lexicographic, and dynamic network flows, *Operations Research* **21**(2) 517-527.

[18] Rebennack S, Arulselvan A, Elefteriadou L and Pardalos PM (2010) Complexity analysis for maximum flow problems with arc reversals, *Journal of Combinatorial Optimization* **19**(2) 200--216.

[19] Ruzika S, Sperber H and Steiner M (2011) Earliest arrival flows on series--parallel graphs, *Networks* **57**(2) 169--173.

[20] Schmidt M and Skutella M (2014) Earliest arrival flows in networks with multiple sinks, *Discrete Applied Mathematics* **164** 320--327.

[21] Wilkinson WL (1971) An algorithm for universal maximal dynamic flows in a network, *Operations Research* **19**(7) 1602-1612.

Hindawi
International Journal of Mathematics and Mathematical Sciences
Volume 2021, Article ID 6651135, 7 pages
https://doi.org/10.1155/2021/6651135

Research Article

Lexicographically Maximum Contraflow Problem with Vertex Capacities

Phanindra Prasad Bhandari and Shree Ram Khadka

Central Department of Mathematics, Tribhuvan University Kathmandu, Kirtipur, Nepal

Correspondence should be addressed to Shree Ram Khadka; shree.khadka@cdmath.tu.edu.np

Received 26 November 2020; Revised 22 January 2021; Accepted 1 February 2021; Published 13 February 2021

Academic Editor: Chin-Chia Wu

The contraflow approach has been extensively considered in the literature for modeling evacuations and has been claimed, due to its lane-direction-reversal capability, as an efficient idea to speed up the evacuation process. This paper considers the contraflow evacuation model on network with prioritized capacitated vertices that allows evacuees to be held at intermediate spots too, respecting their capacities and priority order. In particular, it studies the maximum flow evacuation planning problem and proposes polynomial and pseudo-polynomial time solution algorithms for static network and dynamic multinetwork, respectively. A real dataset of Kathmandu road network with evacuation spaces is considered to implement the algorithm designed for dynamic multinetwork and to observe its computational performance.

1. Introduction

The contraflow approach, which refers to the reversibility of direction of traffic flow in one or more lanes of roadways for fixed time period, re-configures the road network, identifying ideal direction and reallocating available capacity for each arc. The approach, due to its lane-direction-reversal capability, can be taken as a potential remedy to mitigate congestion during emergencies. It significantly reduces the total evacuation time and/or increases the number of evacuees sent from the risk zone to safety. Studies show that reversing one lane of a four-lane dual highway increases the evacuation road capacity by approximately 30%, and by reversing all the inbound lanes, it increases by 67% [1]. The contraflow approach is primarily important for emergency evacuations; nonetheless, its applications are not limited to these. This is commonly used for accommodating directionally imbalanced traffic associated with daily commuter in big cities as well as consequences due to religious gathering, arrangement of concerts or tournaments, etc. However, there is limited implementation of it in real emergency evacuations due to difficulty in using commonly employed methods to duplicate traffic conditions of real contraflow lane during an emergency [1].

The first mathematical optimization model for the contraflow problem was proposed by Rebennack et al. [2] that relies on the basis of the network flow model in [3]. They have investigated analytical solutions for the maximum static contraflow (MSCF) problem and maximum dynamic contraflow (MDCF) problem with polynomial time complexities. The solution idea is based on transformation of input network into a new network for which existing network flow algorithms are applicable. The authors in [4] studied the continuous time maximum dynamic contraflow evacuation problem and proposed a polynomial time solution using the notion of natural transformation of flows suggested in [5].

Other variants that are closely related to the MDCF problem are the quickest contraflow (QCF) problem and earliest arrival contraflow (EACF) problem. The QCF problem on single-source-single-sink network has been solved polynomially in [2]. The EACF problem for the two-terminal series-parallel (TTSP) network has been studied and a polynomial time solution for this has been proposed in [6]. Maximum as well as earliest version of evacuation contraflow problems in network with not necessarily equal transit time on anti-parallel lanes have been studied in [7]. Network reconstruction-based solution procedures have also been proposed for these problems modeled with discrete as well as continuous time setting. The authors in [8] studied these problems for multinetwork setup and proposed polynomial time solutions for both discrete as well as continuous time models. However, the solution procedures for earliest version of the problems work only for TTSP network. The contraflow approach has been incorporated in the network flow model to study facility location problem in [9], and the notion of abstract flow has been applied to network contraflow problems in [10]. The partial contraflow approach over the abstract network setting has been introduced in [11]. We refer to the survey articles [12, 13] for broader insight into dynamic network flow problems and evacuation planning problems.

This paper introduces a new aspect of the evacuation model designed on network with capacitated vertices by imposing the contraflow approach on it. The new model has arc reversal capability and is capable of holding evacuees at temporary shelters at intermediate vertices of given priority. The flow model adopted here is based on weak-conservation constraints given in [14] (cf. [15]). Based on this aspect, the maximum static contraflow problem on ordinary network and maximum dynamic contraflow problem on multinetwork are studied, and solution algorithms for them are proposed. It is crucial, in case of uneven road architecture, for example, to take contraflow models on multinetwork into account for preparing evacuation tasks [8]. Multinetworks capture the situation of road topology with parallel lanes of different transit time and anti-parallel lanes of unequal to and fro transit time. It is considered that the transit time parameter behaves symmetrically during the reversal of arc direction in the case of the dynamic contraflow problem.

The evacuation flow model introduced in [14] is revisited and the lexicographically maximum contraflow problem on network with capacitated vertices is introduced in Section 2. The solution procedures to the problems for static and dynamic cases are proposed in Sections 3.1 and 3.2, respectively. A case illustration with a real dataset is made in Section 4. Section 5 concludes the paper.

2. Model Description

Consider a directed multigraph $G = (V, A)$ with vertex set V and arc set A, both to be finite, such that $n{:}\ = |V|$ and $m{:}\ = |A|$. Represent the source and the sink by s and d, respectively, and assume a terminal set $S \subset V$ with $\mathcal{S}{:}\ = \{v_1, \ldots, v_r\}$ prioritized from higher to lower priority, i.e., $d = v_1 \succ \cdots \succ v_r$, to be given. Then, the corresponding two-terminal evacuation network for time horizon T is represented as $\mathcal{N} = (G, l(a), u(a), \tau(a), k(v), s, d, T)$. Here, $l{:}\ A \longrightarrow \mathbb{N}_0{:}\ = \mathbb{N} \cup \{0\}$ and $u{:}\ A \longrightarrow \mathbb{N}_0$ represent the lower and upper arc capacity functions which bound the number of flow units on each arc $a \in A$ at each time step from below and from above, respectively. Similarly, the vertex capacity function $k{:}\ \mathcal{S} \longrightarrow \mathbb{N}_0$ delimits the total number of flow units, which may be held in each of the vertices $v \in \mathcal{S}$. Moreover, the transit time function $\tau{:}\ A \longrightarrow \mathbb{N}$ specifies the time needed by a flow unit to traverse an arc. Treat time parameter in a discrete manner, i.e., $\mathcal{T}{:}\ = \{0, 1, \ldots, T\}$.

The non-negative flow variables $f(a, t)$ defined by $f{:}\ A \times \mathcal{T} \longrightarrow \mathbb{N}_0$ that specify the flow over time in the network $\mathcal{N}$ are the number of flow units entering arc a at time step t. The number of flow units entering arc a at time step t is assumed to be bounded by the capacity of an arc, i.e., $f(a, t)$ satisfies the capacity constraints for all $a \in A$ and for all $t \in \mathcal{T}$. That is,

$$0 \leq f(a, t) \leq u(a), \quad \forall a \in A, \forall t \in \mathcal{T}. \tag{1}$$

Moreover, $f(a, t)$ has to be equal to zero for all $t > T - \tau(a)$ and for all $a \in A$. The excess flow at vertex $v \in V$ at time $t \in \mathcal{T}$, denoted by $\mathrm{ex}_f(v, t)$, is defined as

$$0 \leq \mathrm{ex}_f(v, t) \coloneqq \sum_{a \in \delta^-(v)} \sum_{\xi=0}^{t-\tau(a)} f(a, \xi) - \sum_{a \in \delta^+(v)} \sum_{\xi=0}^{t} f(a, \xi), \tag{2}$$

where $\delta^-(v){:}\ = \{a \in A{:}\ a = (w, v) \text{for some vertex} \, w \in V\}$ and $\delta^+(v){:}\ = \{a \in A{:}\ a = (v, w) \text{for some vertex} \, w \in V\}$ denote the set of arcs entering and leaving vertex $v \in V$, respectively.

Further, we need to ensure that the excess flow at each vertex $v \in \mathcal{S}$ over time horizon T is to be bounded by the capacity $k(v)$, i.e.,

$$\mathrm{ex}_f(v, T) \leq k(v), \quad \text{for all } v \in \mathcal{S}. \tag{3}$$

Consequently, the total flow of evacuees leaving the source s is equal to the total flow of the evacuees held at vertices $v \in \mathcal{S}$ over the time horizon T, i.e.,

$$\sum_{a \in \delta^+(s)} \sum_{\xi=0}^{T} f(a, \xi) - \sum_{a \in \delta^-(s)} \sum_{\xi=0}^{T} f(a, \xi) = \sum_{v \in \mathcal{S}} \mathrm{ex}_f(v, T). \tag{4}$$

An arc $a = (v, w) \in A$ in which the flow could travel from vertex v to vertex w is replaced by the arc (w, v) for contraflow purpose. The important feature of the considered dynamic network $\mathcal{N}$ is that the capacities and the transit time on anti-parallel arcs could be unequal, and it is allowed to have parallel arcs with different transit time only. Thus, the static network $\mathcal{N}$, we consider here, is not a multinetwork. To this end, the objective of maximum contraflow evacuation planning problem is to lexicographically maximize the vector $(\mathrm{ex}_f(v_1, T), \ldots, \mathrm{ex}_f(v_r, T))^\top$ such that $\mathrm{ex}_f(v_i, T) \leq k(v_i)$ for $i = 1, \ldots, r$, if the direction of arcs on $\mathcal{N}$ is allowed to reverse. The network flow problem with this objective is termed as *lexicographically maximum dynamic contraflow problem* and abbreviated as LexMDCF problem. The maximum contraflow problem with above objective for static network $\mathcal{N} = (G, l(a), u(a), k(v), s, d)$ is termed as *lexicographically maximum static contraflow problem* and is abbreviated as LexMSCF problem.

3. Solution Discussion

Rebennack et al. [2] proposed polynomial time analytical solutions to the MSCF problem and the MDCF problem for the first time. They considered the problems in ordinary network that do not have capability of holding flows at intermediate vertices. Their solution idea is based on the reconstruction of input network into a new one, in which the existing network flow algorithms are applicable. This section discusses the solution procedures to LexMSCF problem for ordinary network and LexMDCF problem for multinetwork based on network reconstruction idea.

3.1. Lexicographically Maximum Static Contraflow Problem. Consider a static network $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(v), s, d)$ with terminal set $\mathcal{S} \subset V$ as described in Section 2. Moreover, consider that $k(d) = \infty$, and consider $k(v)$ to be finite for all $v \in S \backslash \{d\}$. The lexicographically maximum static flow (LexMSF) problem that lexicographically maximizes the amount of flow entering a set of terminals in $\mathcal{S}$ with respect to a given prioritization and given vertex capacities has been solved polynomially in [14]. Here, the objective of the LexMSCF problem is to solve the LexMSF problem on $\mathcal{N}$, if direction of arcs on $\mathcal{N}$ can be reversed.

We modify the solution idea of Rebennack et al. [2] that solves MSCF problem to solve the LexMSCF problem. Their idea is based on modification of input network into a new network by summing the capacities on arcs (v, w) and (w, v) such that MSCF problem reduces to MSF problem on it. In particular, the procedure has following steps. At first, given static network $\mathcal{N} = (V, A, u(a))$ is transformed into its auxiliary network $\widetilde{\mathcal{N}} = (V, \widetilde{A}, u(\widetilde{a}))$ where the arc set $\widetilde{A}$ contains undirected arc (v, w), if (v, w) and/or (w, v) belong to original arc set A with capacity $u(\widetilde{v, w}) = u(v, w) + u(w, v)$. Secondly, a maximum static $s - d$ flow is computed on so-formed undirected network $\widetilde{\mathcal{N}}$ by using any known algorithm. In our case, to ensure the intermediate holding capability in the solution and to respect the vertex capacities, the lexicographically maximum flow is computed by using LexMSF flow computation idea given in [14] instead of computing ordinary maximum flow. The modified procedure that solves LexMSCF problem is given in Algorithm 1.

We state the following lemma that shows equivalence between the optimal flow on $\widetilde{\mathcal{N}}$ and the optimal contraflow on input network $\mathcal{N}$, which turns out to be useful in optimality proof of algorithms designed for contraflow problems in this paper.

Lemma 1 (see [2]). *The maximum static contraflow on a static network $\mathcal{N}$ is equivalent to the maximum static flow on the corresponding transformed network $\widetilde{\mathcal{N}}$.*

Theorem 1. *Given a static network $\mathcal{N} = (V, A, l(a), u(a), k(v), s, d)$, source s and terminal set $\mathcal{S} = \{v_1, \ldots, v_r\} \subset V$ with $d = v_1 \succ \cdots \succ v_r$, and $l(a) = 0$ for all $a \in A$. Then, Algorithm 1 computes a lexicographically maximum static contraflow on $\mathcal{N}$ optimally in strongly polynomial time.*

Proof. The LexMSF Algorithm optimally computes a static flow for each terminals $v \in \mathcal{S}$ as sinks on reduced network $\widetilde{\mathcal{N}}$ (see [14]). Moreover, Lemma 1 shows that these flows are equivalent to the maximum static contraflows on the input network $\mathcal{N}$.

The computational complexity of the algorithm depends on time complexity of the solution procedure on the reduced network $\widetilde{\mathcal{N}}$. This is dominated by the time complexity of the solution procedure of LexMSF problem on $\widetilde{\mathcal{N}}$ since the flow decomposition in each iteration and network transformation can be done only in $O(mn)$, see [16], and $O(m)$ time, respectively. Note that the LexMSF problem can be solved in strongly polynomial time [14]. □

3.2. Lexicographically Maximum Dynamic Contraflow Problem for Multinetwork. Multinetworks capture the evacuation situation with anti-parallel lanes of unequal to and fro transit time as well as parallel lanes of unequal transit time. Maximum dynamic contraflow problems modeled on these class of network without capacitated vertices have been studied in [8]. For given dynamic multinetwork $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(a), s, d, T)$ and terminal set $\mathcal{S}: = \{v_1, \ldots, v_r\}$ of capacitated vertices with $d = v_1 \succ \cdots \succ v_r$, the aim is to solve the LexMDCF problem, if the arc reversibility is permitted only once at time zero. In the following, the solution procedure (Algorithm 2) that solves the MDCF problem for multinetwork is modified to solve LexMDCF problem.

Solving MDCF problem, the arc $(w, v) \in A$ is reversed, if the flow along arc (v, w) exceeds $u(v, w)$ for $\tau(v, w) \leq \tau(w, v)$, or $\tau(w, v) < \tau(v, w)$. This can be viewed, alternatively, as follows: for $(v, w), (w, v) \in A$ such that $\tau(v, w) = \tau(w, v)$, the flow value at arc $\widetilde{a} = (v, w) \in \widetilde{A}$ greater than the capacity $u(v, w)$ of the corresponding arc $(v, w) \in A$ means there is flipping of the direction of arc $(w, v) \in A$. Similarly, in the case with unequal transit time, we can see the sense of flipping the direction of arc $a = (w, v) \in A$, if there is some positive flow on the corresponding arc $\widetilde{a} = (v, w)$. The minimum cost flow (MCF) algorithm applied to generate a dynamic temporally repeated flow ensures that there is flow along the arc (v, w) with less or equal transit time in comparison to the transit time of corresponding anti-parallel arc (w, v), regardless of whether the arc (w, v) is saturated. The parallel arcs $(v, w) \in \widetilde{N}$ have been labeled as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$, for $i = 1, 2, \ldots, q; q \leq m$, to avoid obstruction on the multinetwork while applying MCF algorithm.

Following theorems (Theorems 2 and 3) show that the Algorithm 2 solves maximum dynamic contraflow problem for multinetwork optimally in strongly polynomial time.

Theorem 2. *Given a dynamic multinetwork $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs. Then, maximum dynamic flow on $\widetilde{\mathcal{N}}$ is equivalent to a maximum dynamic contraflow on $\mathcal{N}$.*

Proof. The auxiliary network $\widetilde{\mathcal{N}}$ of the original network $\mathcal{N}$ obtained in step 2 is an undirected multinetwork. The maximum dynamic contraflow problem on $\mathcal{N}$ can be viewed as a maximum dynamic flow problem on $\widetilde{\mathcal{N}}$. While solving the latter problem on $\widetilde{\mathcal{N}}$, the network is to be further transformed by replacing each undirected arc by two oppositely directed arcs with capacities and transit times of both arcs equal to that of original arc. This allows us to send flow on either direction of the arc. However, the flow direction, once chosen, remains fixed throughout the procedure. That is, there is only a flow on one direction of any arc, and never in both directions at the same time as well as at different time periods. However, there could be a flow along arc (v, w) and (w, v) such that $\tau(v, w) \neq \tau(w, v)$ for $(v, w), (w, v) \in A$ at the same time or at different time periods. The latter situation does not make the flow on $\widetilde{\mathcal{N}}$ an infeasible since, in fact, arcs (v, w) and (w, v) are physically different arcs for $\tau(v, w) \neq \tau(w, v)$, due to the labeling of arcs in step 3. Thus, the flow constructed by Algorithm 2 is feasible.

Since every feasible flow of the maximum dynamic flow problem on the transformed network $\widetilde{\mathcal{N}}$ is feasible to the maximum dynamic contraflow problem on network $\mathcal{N}$, the maximum dynamic flow on $\widetilde{\mathcal{N}}$ is not greater than the maximum dynamic contraflow on $\mathcal{N}$. On the other hand, since maximum dynamic flow on network $\mathcal{N}$ does not exceed maximum flow for the corresponding time expanded network $\mathcal{N}^T$ [3], the maximum dynamic contraflow on $\mathcal{N}$ is not greater than the maximum static contraflow in time expanded network $\mathcal{N}^T$. This static contraflow is equivalent to the optimal static flow in $\widetilde{\mathcal{N}}^T$ due to the fact that any maximum static contraflow on network N has equivalent maximum flow in the corresponding transformed network $\widetilde{\mathcal{N}}$ [2]. Again, since there exists a temporally repeated flow which is maximal over the time horizon T [3], the optimal static flow in $\widetilde{\mathcal{N}}^T$ is equivalent to the temporally repeated flow on $\widetilde{\mathcal{N}}$. Thus, the optimal dynamic contraflow on $\mathcal{N}$ is not greater than the optimal dynamic flow on $\widetilde{\mathcal{N}}$. □

Theorem 3. *For dynamic multinetwork $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with integer inputs, Algorithm 2 runs in strongly polynomial time.*

Proof. Construction of auxiliary network in step 2 and labeling parallel arcs in step 3 require only linear time on m. The running time of Algorithm 2 is dominated by computation of a maximum dynamic flow in step 3. It is computed with the help of temporally repeated flow on $\widetilde{\mathcal{N}}$. Finding a temporally repeated flow is equivalent to solving a minimum cost flow problem. The minimum mean cycle-canceling algorithm of [17], for instance, requires $O(n^2 m^3 \log n)$ time for solving this problem. Next effort is to decompose the maximum static flow which requires $O(mn)$ time [16]. Thus, Algorithm 2 runs in a strongly polynomial time for dynamic multinetwork $\mathcal{N}$.

For given dynamic network $\mathcal{N}$ and terminal set $\mathcal{S}$ as described in Section 2, a lexicographically maximum dynamic flow (LexMDF) problem that lexicographically maximizes the amount of flow entering a set of terminals in $\mathcal{S}$ with respect to a given prioritization and given vertex capacities has been studied in [14]. The solution procedure to solve this problem is based on the notion of time expanded network introduced in [3]. Since the objective of LexMDCF problem is to respect the vertex capacities on the prioritized terminals, it is not sufficient to compute a maximum dynamic flow by the means of ordinary temporally repeated flows as in Algorithm 2. Instead, to solve the LexMDCF problem, a lexicographically maximum dynamic flow computation technique proposed in [14] can be applied. The solution procedure to solve LexMDCF problem for multinetwork has been summarized in Algorithm 3. □

Theorem 4. *Given a multinetwork $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(a), s, d, T)$, source s, and terminal set $\mathcal{S} = \{v_1, \ldots, v_r\}$ with $d = v_1 \succ \cdots \succ v_r$, and $l(a) = 0$ for all $a \in A$. Then, Algorithm 3 computes a lexicographically maximum dynamic contraflow on $\mathcal{N}$ in pseudo-polynomial time.*

Proof. The LexMDF algorithm of [14] is applied in Algorithm 3 that computes dynamic flows on transformed network $\widetilde{\mathcal{N}}$ iteratively for each vertex $v \in \mathcal{S}$ in priority order optimally. These flows have equivalent maximum dynamic contraflows on input network $\mathcal{N}$ for each iteration due to Theorem 2. Also, the application of LexMDF algorithm is dominating step with run time depending upon parameter T. Rest of the steps can be performed in strongly polynomial time. Thus, Algorithm 3 computes a lexicographically maximum dynamic contraflow on multinetwork $\mathcal{N}$ in pseudo-polynomial time. □

4. Case Illustration

A case illustration is made by considering Kathmandu road network within and on the Ring Road (see Figure 1). Two scenarios: Scenario I with 38 vertices, 118 arcs, and 13 intermediate shelters and Scenario II (including minor road segments) with 52 vertices, 180 arcs, and 14 intermediate shelters, are examined. New Road area, a highly congested business hub with narrow streets, is taken as the source. Evacuation spaces identified in [18] are taken as the sink (Tribhuvan University (TU) area and Bagmati Corridor near Balkhu) and other intermediate shelters. Standard area (population per 45 square meters sphere) has been considered for the holding capacity of each intermediate evacuation space. However, the sink is assumed to have sufficient capacity. Tables 1 and 2 show the name of evacuation spaces together with their corresponding holding capacities and priority order. For this case illustration, shelters (except sink) are prioritized in random selection (however, it can be done with respect to their capacities, distance from the source or available facilities, and so on). Being a discrete time auto-based evacuation planning model, it followed the "two second rule," considering each minute as

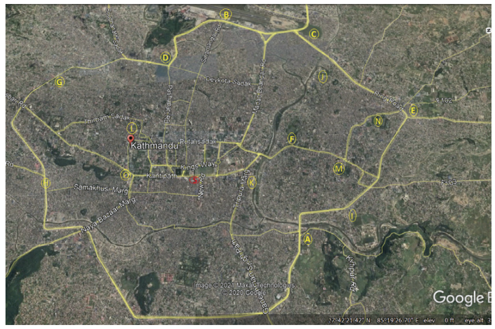


Figure 1: Road network for case illustration taken from Google Earth Pro where S denotes source (New Road) and letters from A to O are sink (Balkhu) and other prioritized intermediate vertices as mentioned in Table 2. Outer boundary traced yellow is Ring Road.

(1) Given a static network $\mathcal{N} = (V, A, l(a), u(a), k(v), s, d)$, $\mathcal{S}: = \{v_1, \ldots, v_r\} \subset V$ with $d = v_1 \succ \cdots \succ v_r$, $l(a) = 0$ for all $a \in A$ and integer inputs.
(2) Transform $\mathcal{N}$ into network $\widetilde{\mathcal{N}} = (V, \widetilde{A}, l(\widetilde{a}), u(\widetilde{a}), k(v), s, d)$ as in [2] and set $l(\widetilde{a}) = 0$, $\forall \widetilde{a} \in \widetilde{A}$.
(3) Solve LexMSF problem on network $\widetilde{\mathcal{N}}$ using algorithm in [14].
(4) Perform flow decomposition into path and cycle flows of maximum flows obtained from step 3 and remove all cycle flows.
(5) Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
(6) Obtain LexMSCF solution on $\mathcal{N}$.

Algorithm 1: LexMSCF algorithm.

(1) Given a multinetwork $\mathcal{N} = (V, A, u(a), \tau(a), s, d, T)$ with single source s, single sink d, and integer inputs.
(2) Transform $\mathcal{N}$ into undirected multinetwork $\widetilde{\mathcal{N}} = (V, \widetilde{A}, u(\widetilde{a}), \tau(\widetilde{a}), s, d, T)$ where $\widetilde{a} = (v, w) \in \widetilde{A}$, if $(v, w), (w, v) \in A$ such that $\tau(v, w) = \tau(w, v)$, with $u(\widetilde{a}) = u(v, w) + u(w, v)$ and $\tau(\widetilde{a}) = \tau(v, w)$, and $\widetilde{a} = (v, w) \in \widetilde{A}$, if $(v, w) \in A$ and $(w, v) \notin A$ such that $\tau(v, w) = \tau(w, v)$, with $u(\widetilde{a}) = u(v, w)$ and $\tau(\widetilde{a}) = \tau(v, w)$.
(3) Label parallel arcs $(v, w) \in \widetilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$, for $i = 1, 2, \ldots, q; q \le m$.
(4) Generate a dynamic, temporally repeated flow on network $\widetilde{\mathcal{N}}$.
(5) Perform flow decomposition into path and cycle flows of the flow resulting from step 4. Remove the cycle flows.
(6) Arc $(w, v) \in A$ is reversed, if and only if the flow along arc (v, w) is greater than $u(v, w)$, or if there is a non-negative flow along arc $(v, w) \in A$.
(7) Get a maximum dynamic contraflow on $\mathcal{N}$.

Algorithm 2: MDCF algorithm [8].

a unit of time. It is considered that the average speed of cars is 550 meters per minute that highly matches with the transit time to travel the segment provided by Google Maps data during normal traffic. Considering the time horizon of 60, 90, and 100 minutes and sufficient sink capacity, the results for Scenario I (Table 1) and Scenario II (Table 2) before and after the application of the contraflow approach are demonstrated.

The results show that the total maximum flow could be increased, while the contraflow approach is applied, by up to approximately 109%, justifying the importance of the approach in the evacuation planning problem. For the real-world problem, it is necessary to restrict the sink also by its actual evacuation space capacity (24090 evacuees [18]). In this case, evacuees that cannot reach sink due to its capacity are distributed among intermediate shelters respecting corresponding capacities, and more than 58 thousand (only nearly 4 thousand less) evacuees in total can be evacuated in 100 minutes for Scenario II, if contraflow is applied. Additional evacuees, about 34 thousand in number for Scenario II with actual sink capacity, can be saved due to consideration of intermediate holding of flows in the evacuation model. Detailed result discussion for this case has been omitted here.

The algorithm has been coded into Python with version 3.9.1 and was run on the computer having Windows 10 operating system with 64 GB RAM and 3.60 GHZ Intel Core i9-9900k processor. It took around 1.5, 6, and 9 minutes to run the program while computing lexicographically maximum flows for $T = 60$, $T = 90$, and $T = 100$, respectively, for Scenario I. Similarly, for Scenario II, it took around 4, 16, and 22 minutes for $T = 60$, $T = 90$, and $T = 100$, respectively. These significantly different running times for different values of parameter T and network input size justify the assertion about time complexity (pseudo-polynomial) of Algorithm 3 made in Theorem 4. The finer discretization of time (controlling parameter in algorithm), instead of considering a

(1) Given a dynamic multinetwork $\mathcal{N} = (V, A, l(a), u(a), \tau(a), k(a), s, d, T)$, $\mathcal{S} = \{v_1, \ldots, v_r\}$ with $d = v_1 \succ \cdots \succ v_r$, $l(a) = 0$ for all $a \in A$ and integer inputs.
(2) Transform $\mathcal{N}$ into undirected multinetwork $\widetilde{\mathcal{N}} = (V, \widetilde{A}, l(\widetilde{a}), u(\widetilde{a}), \tau(\widetilde{a}), k(v), s, d, T)$ as in Algorithm 2 and set $l(\widetilde{a}) = 0$, $\forall \widetilde{a} \in \widetilde{A}$.
(3) Label each parallel arcs $(v, w) \in \widetilde{\mathcal{N}}$ as $(v, w)_i$ such that $\tau(v, w)_i < \tau(v, w)_{i+1}$ for $i = 1, 2, \ldots, q$; $q < m$.
(4) Compute LexMDF on network $\widetilde{\mathcal{N}}$ using algorithm in [14].
(5) Perform flow decomposition into path and cycle flows of maximum flows obtained from step 4 and remove all cycle flows.
(6) Arc $(w, v) \in A$ is reversed if and only if the flow along arc $(v, w) \in A$ is greater than $u(v, w)$ or if there is non-negative flow along arc $a \notin A$.
(7) Obtain LexMDCF solution for multinetwork $\mathcal{N}$.

Algorithm 3: LexMDCF algorithm for multinetwork.

Table 1: Maximum flow values (evacuees) at sink (Balkhu) and intermediate vertices for Scenario I.

SN	Evacuation spaces (vertices)	Vertex capacity	Priority order	Flow/contraflow for $T = 60$	Flow/contraflow for $T = 90$	Flow/contraflow for $T = 100$
1	Balkhu (A)	∞	1	10170/13980	17370/22980	19770/25980
2	Airport (B)	5640	2	2880/5640	3780/5640	4080/5640
3	Koteshor (C)	3270	3	150/3270	150/3270	150/3270
4	Gaushala (D)	2220	4	360/2220	360/2220	360/2220
5	Satdobato (E)	7920	5	0/1710	0/7110	0/7920
6	Pulchowk (F)	2760	6	240/600	240/600	240/1590
7	Shankha Park (G)	240	7	240/240	240/240	240/240
8	Oxygenation Park (H)	3270	8	0/0	0/0	0/0
9	Balkumari (I)	4740	9	0/0	0/0	0/0
10	Teku (J)	2400	10	240/540	240/540	240/540
11	Naxal (K)	900	11	690/900	690/900	690/900
12	Jawalakhel (L)	1440	12	0/0	0/0	0/0
13	Lagankhel (M)	330	13	0/0	0/0	0/0
14	Lainchaur (N)	2130	14	330/2130	330/2130	330/2130

Table 2: Maximum flow values (evacuees) at sink (Balkhu) and intermediate vertices for Scenario II.

SN	Evacuation spaces (vertices)	Vertex capacity	Priority order	Flow/contraflow for $T = 60$	Flow/contraflow for $T = 90$	Flow/contraflow for $T = 100$
1	Balkhu (A)	∞	1	13620/23010	22620/37410	25620/42210
2	Airport (B)	5640	2	2730/5640	3630/5640	3930/5640
3	Koteshor (C)	3270	3	120/2190	120/3270	120/3270
4	Gaushala (D)	2220	4	360/2220	360/2220	360/2220
5	Satdobato (E)	7920	5	0/0	0/2520	0/3660
6	Pulchowk (F)	2760	6	150/480	150/480	150/540
7	Shankha Park (G)	240	7	240/240	240/240	240/240
8	Ring Road (Balaju NG Chowk) (H)	3270	8	690/1470	690/1470	690/1470
9	Oxygenation Park (I)	1470	9	0/0	0/0	0/0
10	Chyasal (J)	4740	10	0/0	0/0	0/0
11	Teku (K)	2400	11	210/420	210/420	210/420
12	Naxal (L)	900	12	210/900	210/900	210/900
13	Jawalakhel (M)	1440	13	0/0	0/0	0/0
14	Lagankhel (N)	330	14	0/0	0/0	0/0
15	Lainchaur (O)	2130	15	180/2130	180/2130	180/2130

minute as a unit of time, would minimize the errors associated with transit time of road segments. However, it leads to higher time complexity to run the program.

5. Conclusion

Importance and applicability of the idea of contraflow especially in the evacuation planning problem has been increasing. Existing network contraflow models fail to capture the situation where it is possible to send evacuees out from the risk zone to even an intermediate capacitated spot if they cannot reach the destination. This paper considered contraflow evacuation planning problems adopting the weak-conservation constraints that allow holding of flow units at prioritized intermediate vertices of given capacities. In particular, it proposed solution algorithms for lexicographically maximum static contraflow problem and lexicographically maximum (discrete) dynamic contraflow problem for multinetwork.

The continuous time dynamic flow model with weak-conservation constraint can also be defined in similar way as given in Section 2 by integrating the flow units at each time point $t \in [0, T)$ instead of adding flow over time for each time step $t \in \mathcal{T}$. Dynamic version of lexicographically contraflow problems studied in this paper can be extended to one with continuous time setting too. These problems can be solved with computational time complexity equal to that of discrete time setting by applying notion of natural transformation of flows suggested in [5].

The limitation of the solution algorithm proposed for dynamic version of the problem is that it leads to a pseudo-polynomial time complexity since the size of underlying network strongly depends on T. Searching of polynomial time solution algorithm for this problem as well as studying the quickest and earliest versions of the problem would be future research areas. One could incorporate the idea of intermediate holding of flows in the multicommodity flow model that better reflects the real-world vehicle distribution scenario of most underdeveloped cities.

Data Availability

The data used for the case illustration are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Phanindra Prasad Bhandari would like to thank the University Grants Commission, Nepal, for providing partial financial support through a PhD Fellowship Award (2016) (Award No. PhD-72/73-S&T-01) to conduct this research.

References

[1] L. Wei, J. Xu, T. Lei, M. Li, X. Liu, and H. Li, "Simulation and experimental analyses of microscopic traffic characteristics under a contraflow strategy," *Applied Sciences*, vol. 9, no. 13, p. 2651, 2019.

[2] S. Rebennack, A. Arulselvan, L. Elefteriadou, and P. M. Pardalos, "Complexity analysis for maximum flow problems with arc reversals," *Journal of Combinatorial Optimization*, vol. 19, no. 2, pp. 200–216, 2010.

[3] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows," *Operations Research*, vol. 6, no. 3, pp. 419–433, 1958.

[4] S. R. Khadka and P. P. Bhandari, "Dynamic network contraflow evacuation planning problem with continuous time approach," *International Journal of Operations Research*, vol. 14, no. 1, pp. 27–34, 2017.

[5] L. Fleischer and É. Tardos, "Efficient continuous-time dynamic network flow algorithms," *Operations Research Letters*, vol. 23, no. 3-5, pp. 71–80, 1998.

[6] T. N. Dhamala and U. Pyakurel, "Earliest arrival contraflow problem on series-parallel graphs," *International Journal of Operations Research*, vol. 10, no. 1, pp. 1–13, 2013.

[7] P. P. Bhandari and S. R. Khadka, "Evacuation contraflow problems with not necessarily equal transit time on anti-parallel arcs," *American Journal of Applied Mathematics*, vol. 8, no. 4, pp. 230–235, 2020.

[8] P. P. Bhandari and S. R. Khadka, *Maximum Contraflow Evacuation Planning Problems on Multi-Network*, (Submitted), 2020.

[9] R. C. Dhungana and T. N. Dhamala, "Maximum flowloc problems with network reconfiguration," *International Journal of Operations Research*, vol. 16, no. 1, pp. 13–26, 2019.

[10] R. C. Dhungana, U. Pyakurel, and T. N. Dhamala, "Abstract contraflow models and solution procedures for evacuation planning," *Journal of Mathematics Research*, vol. 10, no. 4, pp. 89–100, 2018.

[11] U. Pyakurel, H. N. Nath, and T. N. Dhamala, "Partial contraflow with path reversals for evacuation planning," *Annals of Operations Research*, vol. 283, no. 1-2, pp. 591–612, 2019.

[12] T. N. Dhamala, U. Pyakurel, and S. Dempe, "A critical survey on the network optimization algorithms for evacuation planning problems," *International Journal of Operations Research*, vol. 15, no. 3, pp. 101–133, 2018.

[13] M. Skutella, "An introduction to network flows over time," in *Research Trends in Combinatorial Optimization*Springer, Berlin, Heidelberg, 2009.

[14] P. P. Bhandari, S. R. Khadka, S. Ruzika, and L. E. Schäfer, "Lexicographically maximum dynamic flow with vertex capacities," *Journal of Mathematics and Statistics*, vol. 16, no. 1, pp. 142–147, 2020.

[15] P. P. Bhandari and S. R. Khadka, "Maximum flow evacuation planning problem with non-conservation flow constraints at the intermediate nodes," in *Proceedings of International Conference on Mathematical Optimization*, Beijing, China, April 2019.

[16] R. K. Ahuja, T. L. Magnati, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, NJ, USA, 1993.

[17] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *Journal of the ACM*, vol. 36, no. 4, pp. 873–886, 1989.

[18] International Organization for Migration, *Report on Identification of Open Spaces for Humanitarian Purposes in Kathmandu Valley*, https://reliefweb.int/report/nepal/report-identification-open-spaces-humanitarian-purposes-kathmandu-valley, International Organization for Migration, Grand-Saconnex, Switzerland, 2011.

An Efficient Solution Approach to Non-Conservation Maximum Flow Evacuation Planning Problem

1st Shree Ram Khadka
Central Department of Mathematics
Tribhuvan University
Kathmandu, Nepal)
shreeramkhadka@gmail.com

2nd Phanindra Prasad Bhandari
Central Department of Mathematics
Tribhuvan University
Kathmandu, Nepal
phanindra.maths@gmail.com

***Abstract*—Maximum flow evacuation planning problem efficiently sends a maximum flow, i.e. the evacuees from the initial place, the source to the destination, the sink. The problem based on network with flow conservation at each intermediate node has been extensively studied with several algorithms based on Ford and Fulkerson's labeling approach. The problem does not send the flow out from the source to even an intermediate node if this could not be reached to the sink though the nodes other than the source could be relatively safe. In this paper, we propose formulation of the problem with non-conservation constraint and an efficient algorithm to the problem based on the static and on the dynamic network as well. The problem with this constraint allows to send the evacuees in relatively safe places despite of the sink. Moreover, this approach is advantageous to provide immediate medication to severely injured evacuees, to keep the evacuees who lost their lives on the way at the temporary shelters and also to give priority among the evacuees to send to the sink.**

***Index Terms*—evacuation planning, network flow, disaster management**

I. Introduction

The maximum flow evacuation planning problem sends a maximum number of evacuees from a number of sources to a number of sinks via a number of intermediate places over a time horizon during the evacuation or mitigation of the traffic during rush hours in the crowded urban area. The loss of people and their property could be minimized and that the traffic could be efficiently mitigated with the efficient solution strategy to the problem. The problem formulated on the network has been extensively studied with the constraint that the flow is to be conserved at each intermediate node. Several solution procedures have also been investigated, [1].

The problem which deals with no time of sending a unit of flow from one node to the neighboring one is based on the static network, the network only with constant arc capacity. The Ford and Fulkerson's labeling algorithm [4] on the static case yields a maximum flow. Ford and Fulkerson in [5] extended the problem with transit time, the time a unit of flow takes from a node to reach a neighboring node together with the constant arc capacity. A solution, in which the dynamic network is reduced into a time expanded static network, has been investigated with pseudo-polynomial complexity. Moreover, the solution got improved to be polynomial with the transit time along each arc considered as a cost. This improvement generated the temporary repeated flow to obtain the maximum flow over the time horizon. The solution further got improved using the shortest paths from the source to the sink in the residual network [3] and augmenting all the shortest paths at once in each iteration in a layered sub-network of the residual network [2]. The problem with lexicographic concept has been introduced in [10], [11]. An efficient solution to the general lexicographic case of the problem in polynomial time on dynamic network has been investigated in [7]. The problem with contraflow approach in which the arcs are reversed into one direction between the two neighboring arcs has been investigated to be NP-hard in general case [9], [12]. However, contraflow case of the problem with two terminals have been efficiently solved [12]. The solutions to the problem in continuous time setting have been investigated by Khadka and Bhandari in [8] in polynomial time with arc reversal at time zero and in pseudo-polynomial time with arc reversal at any sub-interval of the time horizon.

The problem with flow conservation does not send the flow out from the source to even an intermediate node if this could not be reached to the sink though the nodes other than the source could be relatively safe while evacuating from the source. We need the problem to be formulated with non-conservation flow constraint in the intermediate nodes and to be solved with algorithm which can be executed in a reasonable time as a solution.

The paper has been organized as follows. Section II describes the maximum flow evacuation planning problem with non- conservation flow constraint. Section III explores efficient solution procedure and the last section concludes the paper.

The work of Phanindra Prasad Bhandari has been supported by the University Grants Commission, Nepal.

II. Problem with non-conservation constraint model

We consider d to be the sink where the evacuees are evacuated from the source s via intermediate places $v, v \notin \{s, d\}$, $v \in V$, the set of all nodes of the network $N = (V, E, c_e, \tau_e, s, d, T)$ designed for the evacuation system. The symbol E represents the set of arcs e, c_e for the arc capacity, τ_e for the arc travel time along e for a unit flow of evacuees over the time horizon T. The arc e with $e \in \delta^+(v)$ leaves the node v and that $e \in \delta^-(v)$ enters the node v.

The function for the evacuation flow defined as $f : E \to R^+ \cup \{0\}$, with flow $f(e)$ along the arc e, $e \in E$ follows the following constraints.

The flow $f(e)$ does not exceed the arc capacity c_e, i.e.

$$0 \leq f(e) \leq c_e, \text{ for all } e \in E. \quad (1)$$

The flow value from the node v to the node w equals the flow value from the node w to the node v, i.e.

$$f(v, w) = -f(w, v), \text{ for all } v, w \in V. \quad (2)$$

The flow may not be conserved in the intermediate nodes, i.e.

$$\sum_{e \in \delta^-(v)} f(e) - \sum_{e \in \delta^+(v)} f(e) \geq 0, \ \forall\, v \in V - \{s, d\}. \quad (3)$$

The objective function of the problem is to send a maximum total flow from the source to the sink, i.e.

$$\text{maximize } \mathbf{f_s} = \sum_{e \in \delta^-(d)} f(e). \quad (4)$$

The problem with flow conservation in the intermediate node sends a maximum evacuees from the source to the sink. The limitation is that even a single extra evacuee is not taken out from the source in this case though the intermediate nodes can be safer than the source node. The problem with non-conservation flow in the intermediate node allows to take such evacuees out of the source.

III. Solution Approach

The problem with the flow conservation constraint has been solved with the Ford and Fulkersion labeling algorithm investigated in [4], [5]. The algorithm restricts the flow that enters an intermediate node must exit from there until it reaches the sink. This leads that the flow which cannot reach the sink should not leave the source. However, the problem with non-conservation flow constraint allows the such flows to take out of the source to keep in the intermediate nodes. This characteristic is important to keep more evacuees away from the source during evacuation. The solution to this problem is based on the pre-flow push algorithm in Goldberg and Tarjan [6] with necessary modification.

The algorithm works on a residual network $N_f = (V, E_f)$ of the network N with respect to the flow f. If an arc $(v, w) \in E$ and $f(v, w) < c_{(v,w)}$, then E_f contains the arc (v, w) with residual capacity $r_{(v,w)} = c_{(v,w)} - f(v, w)$ and if $(v, w) \in E$ and $f(v, w) > 0$, then E_f contains the arc (w, v) with residual capacity $r_{(w,v)} = f(v, w)$. The flow $f(e)$ for the former and the later case are said to be forward flow and backward flow respectively. The nodes are labeled with the label function $l : V \to Z^+ \cup \{0\}$ defined as

$$l(v) \begin{cases} = |V| & \text{if } v = s, \\ \leq l(w) + 1 & \text{if } (v, w) \in N_f, \\ = 0 & \text{if } v = d. \end{cases}$$

The nodes are initially labeled to be $l(s) = |V|$ and $l(v)$ the shortest distance from the sink d for all $v \in V - \{s\}$ by considering the unit length for each arc. The initial flow is set as $f(e) = c_e$ for all $e\delta^+(s)$ and zero otherwise. The algorithm pushes flow units δ from an active node $v \neq s, d$ to its neighboring node $w \neq s$, $(v, w) \in N_f$ when the label at v and w satisfies $l(v) = l(w) + 1$. The node v is active if it contains an excess flow e_v defined as $e_v \leq \sum_{e \in \delta^-(v)} f(e)$ and the flow units to be pushed is $\delta = \min\{e_v, c_e\}$. The node v is relabeled to be $l(v) := 1 + min\{l(w) : (v, w) \in N_f; w \neq s\}$, if $l(v) \leq l(w)$ with $e_v > 0$. The excess flow e_v is held in the temporary shelter v' of the node v even if $l(v) \leq l(s)$ cannot be satisfied after relabeling.

The existing pre-flow push algorithm described in Goldberg and Tarjan [6] sends the excess flow back to the source. However, this modification keeps it in the intermediate nodes. This seems to be applicable to the evacuation planning problem to provide immediate medication to severely injured evacuees, to keep the properties and the evacuees who lost their lives on the way at the temporary shelters and also to priority among the evacuees to send to the sink.

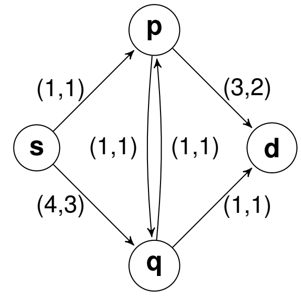


Fig. 1. Network N

Let us consider a network, for example, Figure 1 in which the first component stands for the arc capacity and the second one for the transit time given as the order pairs issued in each arc. If we do not consider the transit time, the network becomes a static network. The algorithm sends a maximum flow from the source s to the sink d and keeps the remaining flow taken out of the source s at a temporary shelter of intermediate nodes. The network becomes a dynamic network if one considers the transit time also. The algorithm can be applied on the static network and on the dynamic network if transferred into a time expanded static network.

The network $N = (V, E, c_e, \tau_e, s, d, T)$ is transferred into the corresponding time expanded network

$N^T = (V^T, E^T, c_e, \tau_e, s, d, T)$ by discretizing the time horizon T to be $\theta \in \{0, 1, ..., T\}$ and the node set V^T and the arc set E^T to be

$$V^T = \{v(\theta) : v \in V \text{ and } \theta \in \{0, 1, ..., T\}\}$$

and

$$E^T = \{(v(\theta), w(\theta + \tau(v, w))) : v \neq w,\ v, w \in V$$

and

$$\theta \in \{0, 1, ..., T - \tau_e\},$$

respectively. It is noteworthy that the Ford and Fulkerson's algorithm also works to obtain the maximum flow to the problem with the flow conservation case on the time expanded network [5].

Consider a static network from the Figure 1. The Ford and Fulkerson's labeling algorithm sends a maximum flow with the value of 3 units of evacuees from the source node s to the sink node d for the flow conservation case of the problem. The proposed algorithm for the non-conservation case of the problem also sends the equal amout of the evacuees to the sink node from the source node. Moreover, this algorithm takes 2 units of evacuees out from the source node s and keeps at a relative safe node q. If we set priority among the intermediate nodes, the excess flow units of the evacuees can be sent to the prioritized relatively safe nodes.

The algorithm can be executed in polynomial time which takes $O(n^2)$ time to relabel the nodes, $O(n^2 m)$ to push the flow units to the neighboring nodes and $O(n^2)$ to hold the excess flow units in the intermediate nodes. The cardinality of the arc and the node sets are $|E| = m$ and $|V| = n$, respectively.

Now we consider the dynamic network N mentioned in Figure 1 with the time horizon $T = 5$. The algorithm sends a maximum flow with the value of 5 units of evacuees from the source to the sink over the time horizon. Moreover, the algorithm keeps $1, 3, 1, 3, 4$ units of evacuees at the intermediate nodes p, q, p, q, q at time $5, 3, 4, 4, 5$, respectively. The maximum units of evacuees held at the intermediate nodes is 12. The algorithm applied for time expanded network is executed in pseudo-polynomial complexity with $O(T^2 n^2)$, $O(T^3 n^2 m)$ and $O(T^2 n^2)$ time to relabel, to push the flow units and to hold, respectively.

IV. Concluding Remarks

The maximum flow evacuation planning problem is useful to save people and their properties during natural and man made disasters using efficient solution procedures which have been investigated in the literature. The problem with flow conservation in the intermediate nodes has been widely studied with several efficient solution strategies. In this paper we have considered the problem in which the flow may not be conserved in the intermediate nodes. An efficient solution that works to solve the problem designed on the static network and that on the dynamic network in which the transit time in the arcs has been proposed. The solution sends a maximum units of flow of evacuees from the dangerous zone to the safety zone. This value is exactly equal to the maximum flow units sent by the solution techniques investigated to solve the problem with flow conservation aspect. Despite of this, the proposed solution sends several units of flow of evacuees to relatively safe zones from the dangerous zone which is not possible using already investigated solution techniques.

The proposed algorithm is the modification of the algorithm investigated in Goldberg and Tarjan [6]. Moreover, it has been extended to the dynamic network using the time expanded graph approach. The computational performance of the algorihms, the maximum flow aspect of the problem with multiple sources and sinks and the other aspects for example the earliest arrival flow, the quickest flow of the problem are the interesting area for further research.

References

[1] T. N. Dhamala, "A survey on models and algorithms for discrete evacuation planning network problems," Journal of Industrial and Management Optimization, vol. 11, pp. 265-289, 2015.

[2] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in networks with power estimation," Soviet Mathematics Doklady vol. 11 pp. 1277-1280, 1970.

[3] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," Journal of Association for Computing Machinery, vol. 19 (2), pp. 248-264, 1972.

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

[5] L. R. Ford and D. R. Fulkerson, "Constructing maximal dynamic flows from static flows," Operations Research, vol. 6, pp. 419-433, 1958.

[6] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," Journal of Association for Computing Machinery, vol. 35, pp. 921-940, 1988.

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

[8] S. R. Khadka and P. P. Bhandari, "Dynamic network contra-flow evacuation planning problem with continuous time approach," International Journal of Operations Research, vol. 14(1), pp. 27-34, 2017.

[9] S. Kim, S. Shekhar and M. Min, "Contra-flow transportation network reconfiguration for evacuation route planning," IEEE Transactions on Knowledge and Data Engineering, vol. 20, pp. 1-15, 2008.

[10] N. Megiddo, "Optimal flows in networks with multiple sources and sinks," Mathematical Programming, vol. 7, pp. 97-107, 1974.

[11] E. Minieka, "Maximal, lexicographic, and dynamic network flows," Operations Research, vol. 21, pp. 517-527, 1973.

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

Contraflow Evacuation Planning Problems for Disaster Management

Phanindra Prasad Bhandari[1] and Shree Ram Khadka[2]

Abstract

An evacuation planning problem gives a plan on existing road network that sends as many evacuees as possible from the dangerous place to the safer place in minimum time period efficiently during disasters. The problems with different road network attributes have been studied and solutions have been proposed. Evacuation planning problems with network contraflow approach, reversing the direction of lanes, have also been extensively studied. In this paper, we study the contraflow problems, namely, maximum dynamic contraflow problems and earliest arrival contraflow problems without necessarily equal transit times on two oppositely oriented arcs joining the same two different nodes and propose efficient solutions to them when arc reversal capability is allowed only once at time zero.

Keywords: *Network Flow; Contraflow; TTSP Network; Evacuation Planning Problem; Disaster Management.*

1. Introduction

An evacuation planning problem, important notion during the response phase of the disaster management, attempts to find an optimal evacuation plan with a realistic flow model where each evacuee is supposed to be evacuated in a minimal time period from a dangerous site (source) to a safe site (sink). An efficient evacuation plan minimizes the human casualties and their property during natural and human-created disasters and helps in mitigation of rush hour traffic in the crowded urban area.

Maximum dynamic flow problem that sends the maximum amount of flow from the source to the sink within the given time horizon was due to Ford and Fulkerson (1958, 1962) and has been extensively studied with various applications. The earliest arrival flow problems that ask to maximize flow into the sink at each time step within the time horizon have also been considered widely in the literature; Gale (1959), Minieka (1973), Wilkinson (1971), Hoppe (1995), Hoppe & Tardos (1994), Baumann (2007), Steiner (2009), and Ruzika *et al.* (2011).

Consideration of contraflow approach in evacuation planning problem could be a potential remedy during disasters due to its lane direction reversal property, see, Dhamala (2015). The contraflow approach reconfigures the network identifying the ideal direction and reallocating the available capacity for each arc to minimize the evacuation time from source to sink. Two heuristic solutions to the problem are proposed in Kim *et al.* (2008). There exists an analytical solution also for the problem which sends a maximum flow from a source to a sink in the two terminal case, see Rebennack *et al.* (2010). The solution with polynomial time complexity has been investigated on both static and dynamic networks where the arc reversal ability has been adapted only once at very beginning of the time horizon for later type of networks. Earliest arrival contraflow problem on two terminal series parallel network has been considered in Dhamala & Pyakurel (2013).

The dynamic network flow problems in continuous time setting can be found in Philpott (1982), Anderson *et al.* (1982), Philpott (1990), Anderson and Philpott (1994), Fleischer & Tardos (1998), Koch *et al.* (2011) and Hashemi & Nasrabadi (2012). The dynamic network contraflow problem in continuous time setting has been introduced in Khadka & Bhandari (2017), and Pyakurel & Dhamala (2017). Continuous time model naturally better reflects the real-world evacuation scenario.

The remaining part of the paper is organized as follows. Basic mathematical denotations related to the problems have been described in Section 2. Section 3 contains formulation of the maximum dynamic contraflow problem with discrete time setting and continuous time setting in Subsection 3.1

[1] Science and Humanities Department, Khwopa Engineering College, Bhaktapur, Nepal, phanindra.maths@gmail.com
[2] Central Department of Mathematics, Tribhuvan University, Kirtipur, Nepal

and Subsection 3.2 respectively. Their solution techniques and that of earliest arrival contraflow problem is described in Subsection 3.3. The last section concludes the paper.

2. Preliminaries

Consider an evacuation network $N = (V,\ E,\ c_e,\ \tau_e,\ T)$ where V is the set of nodes v, the crossing of routes from dangerous place, source s, to safer place, sink d and E is the set of route segments, arc $e = (v, w)$ joining any two different nodes $v, w \in V$. Let $c: E \rightarrow Z^{\geq 0}$ be a capacity function denoting the upper bound for flow units to pass the arc at a time slot and $\tau: E \rightarrow Z^{>0}$ be the transit time function denoting the time required for a flow unit to travel the arc. The capacities and transit times on the arcs joining the same two different nodes with different orientations could be unequal. That is, $c_{\overleftarrow{e}}$ may or may not be equal to $c_{\overrightarrow{e}}$ and $\tau_{\overleftarrow{e}}$ may or may not be equal to $\tau_{\overrightarrow{e}}$ where $\overrightarrow{e}$ and $\overleftarrow{e}$ denotes the arc $e = (v, w) \in E$ with forward direction and backward direction respectively. The total evacuation time period is denoted by T. An evacuation network has been depicted in Fig. 1.

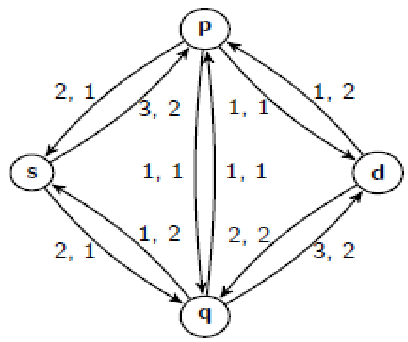


Fig. 1. An evacuation network N. Line with arrow joining any two ovals (the node) is an arc representing the road with direction of the flow, the movement of evacuees. Node s is the dangerous place (source) that contains evacuees, d is the safe place (sink) and the remaining nodes are the intermediate (transshipment) nodes. The first and the second numbers next to each arc are the arc capacity and the transit times respectively.

The flow of evacuees f on N defined as $f: E \times [0, T) \rightarrow R$^$(\geq 0)$ satisfies the following conditions:
The flow units cannot exceed the arc capacity c_e for any time within the time horizon. That is,

$$0 \leq f(e, \theta) \leq c_e \ \forall\ e \in E \text{ and } \forall\ \theta \in [0, T). \quad (1)$$

The flow units that enters into node v for all $v \in V - \{s, d\}$ must exit from it. That is,

$$\sum_{e \in \delta^-(v)} \int_0^T f(e, \theta - \tau_e) d\theta = \sum_{e \in \delta^+(v)} \int_0^T f(e, \theta) d\theta \quad (2)$$

where $\delta^-(v)$ and $e \in \delta^+(v)$ denote for the set of arcs entering into the node v and leaving from it, respectively.

An $s - d$ flow in which the flow function f satisfies constraints (1) and (2) for all arcs at time $\theta \in [0, T)$ is a feasible $s - d$ flow. For a dynamic network $N = (V, E, c_e, \tau_e,\ s, d, T)$, the maximum dynamic $s - d$ flow problem finds the optimal feasible flow from s to d within the given time horizon T.

If one wishes to send packets of flow units at discrete time points into the arcs instead of sending flow at continuous flow rates, the time horizon T is to be discretized into the time steps $\{0, 1, \ldots, T\}$. In this model, the flow units sent into an arc $e = (v, w)$ at a time θ totally reach the target node w at time $\theta + \tau_e$. In discrete time setting, the flow function f defined as $f: E \times \{0, 1, \dots, T\} \rightarrow R^{\geq 0}$ satisfies the capacity constraint in the form:

$$0 \leq f(e,\theta) \leq c_e \; \forall \, e \in E \text{ and } \forall \; \theta \in \times \{0,1,\dots,T\} \quad (3)$$

and the flow conservation constraint in the form:

$$\sum_{e\in\delta^-(v)} \sum_{\theta=0}^{T} f(e,\theta - \tau_e) = \sum_{e\in\delta^+(v)} \sum_{\theta=0}^{T} f(e,\theta) \quad (4)$$

An $s - d$ flow f satisfying capacity constraints (3) and flow conservation constraints (4) for all arcs at every time steps $\theta \in \{0,1,\dots,T\}$ is a feasible $s - d$ flow for discrete time setting.

Obviously, the flow value before and after the time horizon T is zero and all flow units leave the network within it.

3. Contraflow Problems

The problem of finding the arcs to reverse its direction so that the maximum flow on network from the source to the sink could be increased and time required to attain this flow could be decreased is a network contraflow problem. Rebennack et al. (2010) studied the maximum dynamic contraflow problem with equal transit times on two oppositely oriented arcs joining the same two different nodes. Their model is with discrete time setting and arc reversal capability has been allowed only once at time zero.

In this section, we define a maximum dynamic contraflow problem with unequal transit times on the arcs with discrete as well as continuous time settings in which arc reversal capability is allowed only once at time zero.

3.1. Maximum Dynamic Contraflow with Discrete Time Setting

Let $N = (V,\ E,\ c_e,\ \tau_e,\ s,\ d,\ T)$ be the network with single source s and single sink d and transit time τ_e for each arc $e \in E$ with $\tau_{(v,w)}$ not necessarily equal to $\tau_{(w,v)}$ if $(v,w),(w,v) \in E$. The maximum dynamic network flow problem with discrete time setting on N, if the direction of the arcs is allowed to reverse, maximizes the net flow f_d from s to d given by

$$f_d = \sum_{e\in\delta^+(s)} \sum_{\theta=0}^{T} f(e,\theta) - \sum_{e\in\delta^-(s)} \sum_{\theta=0}^{T} f(e,\theta)$$
$$= \sum_{e\in\delta^-(d)} \sum_{\theta=0}^{T} f(e,\theta) - \sum_{e\in\delta^+(d)} \sum_{\theta=0}^{T} f(e,\theta) \quad (5)$$

where the flow f satisfies the capacity constraints (1) and the flow conservation constraints (2).

The solution to the problem is based on the network transformation. In contrast to the case of equal transit times on the arcs, addition of capacities of oppositely directed arcs joining two nodes, while constructing the auxiliary network, is no longer possible in the case of unequal transit times. We propose an alternative, more general, way of constructing the auxiliary network for the latter case. Network N with unequal transit times and capacities on the arcs is transformed into new network by introducing an artificial node for each arc that separates it into two different arcs. Each arc on N is splitted into two arcs; real arc and artificial arc; in the auxiliary network N^+. Artificial arcs have infinite capacities and zero transit times whereas the real arcs have the original arc capacities and transit times. Moreover, corresponding to each arc $e^+ \in N^+$, an oppositely directed artificial arc with zero capacity and transit time equal to τ_{e^+} is inserted, see Fig. 2.

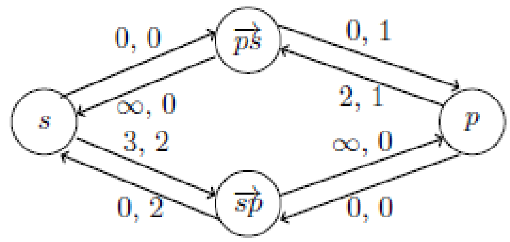


Fig. 2. A part of the transformed network N^+ of the network N depicted in Fig. 1.

Now we propose an efficient (having polynomial time complexity) solution procedure, say Discrete Time Maximum Dynamic Contraflow Algorithm (MDCF-DT), which solves the maximum dynamic contraflow problem with discrete time setting, if the direction of the arcs is allowed to reverse only once at time zero. The procedure resembles with algorithm designed for the discrete model with equal transit times in Rebennack et al. (2010).

Algorithm 1: MDCF-DT

1. *Given network $N = (V, E, c_e, \tau_e, s, d, T)$ with transit time τ_e for each arc $e \in E$ with $\tau_{(v,w)}$ not necessarily equal to $\tau_{(w,v)}$ if $(v, w), (w, v) \in E$.*
2. *Transform network N into N^+.*
3. *Apply the algorithm of Rebennack et al. (2010) on N^+.*
4. *Get the maximum dynamic contraflow with discrete time setting on N for time horizon T.*

Applying Algorithm 1 for the evacuation network N given in Fig 1 with time horizon $T = 10$, the maximum dynamic contraflow with discrete time setting is of value 52 whereas the maximum dynamic flow without contraflow is of value 30.

3.2. Maximum Dynamic Contraflow with Continuous Time Setting

Let $N = (V, E, c_e, \tau_e, s, d, T)$be the network with single source s and single sink d and transit time τ_e for each arc $e \in E$ with $\tau_{(v,w)}$ not necessarily equal to $\tau_{(w,v)}$ if $(v, w), (w, v) \in E$. The maximum dynamic flow problem with continuous time setting on N if the direction of the arcs is allowed to reverse, maximizes the net flow flow $f_c$from s to d given by

$$f_c = \sum_{e\in\delta^{+}(s)} \int_0^T f(e,\theta)d\theta - \sum_{e\in\delta^{-}(s)} \int_0^T f(e,\theta)d\theta$$
$$= \sum_{e\in\delta^{-}(d)} \int_0^T f(e,\theta)d\theta - \sum_{e\in\delta^{+}(d)} \int_0^T f(e,\theta)d\theta \quad (6)$$

where the flow f satisfies the capacity constraints (3) and the flow conservation constraints (4).

Flow value derived with discrete time setting can be used to find the flow with continuous time setting using the notion of natural transformation of flow discussed by Fleischer and Tardos (1998) which states that the amount of flow that arrives at the node v through the arc $\vec{e}$ at time step θ in the discrete approach is equal to the amount of flow arriving at w through the arc $\vec{e}$ during the unit interval of time at the beginning of time step θ i.e. $f_d(e, \theta) = f_c(e, [\theta, \theta + 1))$ for all $\theta \in \{0, 1, \ldots, T - 1\}$. The notion can also be used to transform the time-expanded network constructed with discrete time setting into the time-expanded network with continuous time setting.

Now we propose an efficient (having polynomial time complexity) solution procedure, say Continuous Time Maximum Dynamic Contraflow Algorithm (MDCF-CT), which solves the maximum dynamic contraflow problem with continuous time setting, if the direction of the arcs is allowed to reverse only once at time zero.

Algorithm 2: MDCF-CT

1. *Given network* $N = (V, E, c_e, \tau_e, s, d, T)$ *with transit time* τ_e *for each arc* $e \in E$ *with* $\tau_{(v,w)}$ *not necessarily equal to* $\tau_{(w,v)}$ *if* $(v, w), (w, v) \in E$.
2. *Transform network* N *into* N^+.
3. *Apply the algorithm of Khadka and Bhandari (2017) on* N^+.
4. *Get the maximum dynamic contraflow with continuous time setting on* N *for time horizon* T.

Applying Algorithm 2 for the evacuation network N given in Fig 1 with time horizon $T = 10$, the maximum dynamic contraflow with continuous time setting is of value 45 whereas the maximum dynamic flow without contraflow is of value 26.

3.3. Earliest Arrival Contraflow on TTSP Network

Let $N = (V, E, c_e, \tau_e, s, d, T)$be the network with single source s and single sink d and transit time τ_e for each arc $e \in E$ with $\tau_{(v,w)}$ not necessarily equal to $\tau_{(w,v)}$ if $(v, w), (w, v) \in E$. The earliest arrival contraflow (EACF) problem with discrete time setting, if the direction of the arcs is allowed to reverse, maximizes the feasible net flow from s to d at all time steps $\theta \in \{0, 1, \ldots, T\}$. We consider a special class of network known as two terminal series-parallel (TTSP) network N here. A two-terminal series-parallel network N is a directed network with a single source s and a single sink d which has a single arc (s, d) together with source s and sink d or is obtained from two series parallel networks N_1 and N_2 by one of the following two operations:

(I) **Parallel Composition**: Merge source nodes s_1 of N_1 and s_2 of N_2 to form the source s of N and merge sink nodes d of N_1 and d_2 of N_2 to form the sink d of N.
(II) **Series Composition**: Merge the sink node d_1 of N_1 with the source node s_2 of N_2.

Now we present an efficient (having polynomial time complexity) solution procedure, say Discrete Time Earliest Arrival Contraflow Algorithm (EACF-DT), that is based on the algorithm of Ruzika *et al.* (2011) for minimum cost flows on TTSP networks. The algorithm solves the maximum dynamic flow problem using a temporally repeated flow over the time horizon T. In fact, this maximum dynamic flow has the earliest arrival property. We claim that the two terminal series parallel network N, after transforming into its auxilary network N^+, remains two terminal series parallel network. That is, the following algorithm, Algorithm 3, solves the earliest arrival contraflow problem with discrete time setting, if the direction of the arcs are allowed to reverse only once at time zero.

Algorithm 3: EACF-DT

1. *Given a TTSP network* $N = (V, E, c_e, \tau_e, s, d, T)$ *with transit time* τ_e *for each arc* $e \in E$ *with* $\tau_{(v,w)}$ *not necessarily equal to* $\tau_{(w,v)}$ *if* $(v, w), (w, v) \in E$.
2. *Find auxiliary network* N^+.
3. *Solve earliest arrival flow problem on* N^+ *by using the algorithm of Ruzika et al. (2011).*
4. *Arc* $\overleftarrow{e}$ *is reversed if and only if the flow along arc* $\vec{e}$ *is greater than* c_e*or if there is non-negative flow along arc* $e \notin E$.
5. *Get the earliest arrival contraflow with discrete time setting on* N *for time horizon* T.

Together with the notion of natural transformation, the Algorithm 3 (EACF-DT) solves the earliest arrival contraflow problem with continuous time setting on N when the arc reversal capability is allowed only once at time zero. And, of course, the solution to the problem is optimal and can be found in strongly polynomial time.

4. Conclusion

The importance and applicability of the idea of network contraflow especially in evacuation planning problem has been increasing due to its lane direction reversal capability. The model of the problem without necessarily equal transit times on two oppositely oriented arcs joining the same two different nodes and its solution approaches based on continuous time setting better reflects the real-world situation. In this paper, we studied the maximum dynamic contraflow problem and earliest arrival contraflow problem with strongly polynomial time algorithms as solution procedures if the arcs are flipped only once at time zero in discrete as well as continuous time setting. Implementation of the model to the real data set would be the immediate future research work.

Acknowledgement

The first author would like to thank the University Grants Commission, Nepal for providing partial financial support (PhD Fellowship 2016) to conduct this research.

References

Anderson, E.J., Nash, P., & Philpott, A.B. (1982). A class of continuous network flow problems. *Mathematics of Operations Research*, 7, 501-514.
Anderson, E.J. and Philpott, A.B. (1994). Optimisation of Flows in Networks over Time. Probability, Statistics and Optimisation, F. P. Kelly, ed., Wiley, New York, ch. 27, 369-382.
Baumann, N. (2007). *Evacuation by Earliest Arrival Flows*. PhD thesis, Department of Mathematics, University of Dortmund, Germany.
Dhamala, T.N. (2015). A survey on models and algorithms for discrete evacuation planning network problems. *Journal of Industrial and Management Optimization*, 11, 265--289.
Dhamala, T.N., & Pyakurel, U. (2013). Earliest arrival contraflow problem on series-parallel graphs. *International Journal of Operations Research*, 10, 1--13.
Fleischer, L., & Tardos, E. (1998). Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23, 71--80.
Ford, L.R. and Fulkerson, D.R. (1958). Constructing maximal dynamic flows from static flows. *Operations Research*, 6, 419--433.
Ford, L.R., & Fulkerson, D.R. (1962). *Flows in Networks*. Princeton University Press.
Gale, D. (1959). Transient flows in networks. *Michigan Mathematical Journal*, 6, 59--63.
Hoppe, B. (1995). *Efficient Dynamic Network Flow Algorithms*. PhD thesis, Cornell University.
Hoppe, B., & Tardos, E. (1994). Polynomial time algorithms for some evacuation problems. *Procc. of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 433--441.
Hashemi, S.M., & Nasrabadi, E. (2012). On solving continuous-time dynamic network flows. *Journal of Global Optimization*, 53, 497--524.
Khadka, S.R., & Bhandari, P.P. (2017). Dynamic network contraflow evacuation planning problem with continuous time approach. *International Journal of Operations Research*, 14, 27--34.
Minieka, E. (1973). Maximal, lexicographic, and dynamic network flows. Operations Research, 21, 517--527.
Koch, R., Nasrabadi, E., & Skutella, M. (2011). Continuous and discrete flows over time: a general model based on measure theory. *Mathematical Methods of Operations Research*, 73, 301--337.
Philpott, A.B. (1982). *Algorithms for Continuous Network Flow Problems*. PhD thesis, University of Cambridge, UK.
Philpott, A.B. (1990). Continuous-time flows in networks. *Mathematics of Operations Research*, 15, 640--661.
Pyakurel, U., & Dhamala, T.N. (2017). Continuous dynamic contraflow approach for evacuation planning. *Annals of Operations Research*, 253, 573--598.
Rebennack, S., Arulselvan, A., Elefteriadou, L., & Pardalos, P.M. (2010). Complexity analysis of maximum flow problem with arc reversal. *Journal of Combinatorial Optimization*, 19, 200--216.
Ruzika, S., Sperber, H., & Steiner, M. (2011). Earliest arrival flows on series-parallel graphs. *Networks*, 57, 169--173.
Steiner. M. (2009). *A Survey on Earliest Arrival Flows and a Study of the Series-parallel Case*. Diploma Thesis, Department of Mathematics, University of Kaiserslautern, Germany.
Wilkinson, W.L. (1971). An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19, 1602--1612.

Evacuation Planning Problems with Intermediate Storage

Phanindra Prasad Bhandari* and Shree Ram Khadka

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

* Corresponding author's email: phanindra.maths@gmail.com

doi:https://doi.org/10.21467/proceedings.100.9

ABSTRACT

An evacuation planning problem gives a plan on existing road network for disaster management that attempts to send all evacuees from the dangerous zone to the safer zone efficiently. The network flow problems provide important tools for modeling the evacuation tasks. The problems based on the model with flow conservation constraint, that permits an evacuee to be taken out of the disastrous zone only if it can be sent into the safe zone, have been extensively studied for various evacuation scenarios. In this paper, we study dynamic flow problems based on weak flow conservation constraints that allow for an intermediate node to serve as a temporary shelter also with three distinct objectives and propose efficient solution procedures. The first is to maximize the number of evacuees into the safe zones in priority order within the specified time horizon. The second is to achieve the first objective at every time point within the time horizon. And, the third is to fulfill the demand (number of evacuees) at each of the safe zones in minimum possible time horizon in priority order.

1 Introduction

The maximum flow problem, investigated by L. R. Ford and D. R. Fulkerson in 1950s, is the foundation of all the mathematical optimization based evacuation planning problems. Time plays crucial role in modeling real-world evacuation scenarios. The network flow problem known as *maximum dynamic flow (MaxDF) problem*, aiming to send the maximum number of flow unit from the source into the sink in specified time horizon, has been introduced in [7, 8]. There is a pseudo-polynomial algorithm based on the time-expanded network and a polynomial one based on the temporally repeated flow with transit times on the arc as a cost coefficients to solve MaxDF problem on two terminal network. The network flow problems with continuous time setting have been studied in [14, 6, 1, 15].

A problem closely related to a maximum dynamic flow problem is the *quickest flow (QF) problem* that sends a given units of flow from the source to the sink in minimum possible time. This problem can be solved in polynomial time by incorporating the algorithm to solve a maximum dynamic flow problem in a binary search framework. Using Megiddo's method of parametric search [12], a faster algorithm which solves the quickest flow problem in strongly polynomial time can be obtained [4].

The problem that attempts to send a maximum number of evacuees from the source to the sink as earliest as possible within given time horizon is the *earliest arrival flow (EAF) problem*. Gale [9] introduced EAF problem to obtain the maximum amount of flow for every discretized time steps of evacuation time horizon. There exist exponential-time exact algorithms also for the problem [13, 21]. A solution technique for the problem over two terminal series parallel (TTSP) networks that runs with polynomial time complexity has been proposed in [20, 18].

Contraflow approach, reversing the direction of arcs, has also been considered in evacuation planning problems that increases the outbound capacity of the arc and decreases the evacuation time. Here, arcs represent the lanes of a road within the evacuation zone. The first analytical solutions for the maximum contraflow problem on static as well as dynamic network are due to [17]. The evacuation planning problem with contraflow approach in continuous time setting has been studied in [11] and [16]. For a broader overview on evacuation planning problems, we refer to the survey articles [19] and [5].

All the flow problems discussed so far are based on the model with flow conservation at intermediate nodes for which no evacuee is sent out of the source if it cannot reach the sink. There may be some intermediate nodes over

Proceedings DOI: 10.21467/proceedings.100; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-942709-6-6 (eBook)

evacuation network with holding capacity and are relatively safe as compared with the source which are useful to support more evacuees. This paper considers evacuation planning problems over a network which consists of some prioritized intermediate nodes with given storage capacities. The priority depends on how safe the intermediate place is and/or how much capacity does it have. The flow may not be conserved at such intermediate nodes rather can be held at them. The problem without node capacity can be solved efficiently using the notion of a temporally repeated flows (TRFs) generated by repeating all possible source to sink path flows, see [7, 8]. As far as author know, there is no polynomial time method to compute a temporally repeated flow that solves the problem on general network with limited node capacity at intermediate nodes of given priority order.

We revisit the lexicographic maximum dynamic flow (LexMaxDF) problem introduced in [3] in Section 2 and discuss its solution idea in Section 3. The lexicographic earliest arrival flow (LexEAF) problem and the lexicographic quickest flow (LexQF) problem are introduced and solution procedures to them are proposed in Sections 4 and 5, respectively. Section 6 extends the results in continuous time setting. Section 7 concludes the paper.

2 Model Description

An evacuation scenario is represented by a network $N = (V, A, c(a), k(v), \tau_a, T)$ with $|V| = n$, $|A| = m$ where V is the set of nodes v denoting the crossings of road segments, A the set of arcs $a = (v, w), v, w, \in V$ denoting the road segments, $c(a) : c(a) \in Z^+ \cup \{0\}$ is the arc capacity which is the upper bound for the evacuees to pass along the arc a in a unit time, $k(v) : k(v) \in Z^+ \cup \{0\}$ is the node capacity which is the upper bound of evacuees to be held at node v, τ_a a non-negative integer, the transit time which is the time required for an evacuee to travel along arc a and T is the time horizon within which the evacuation process is supposed to be completed. Special nodes denoted by s and d are the source and the sink, respectively.

For a discrete dynamic flow model, the non-negative flow variables $f : A \times \{0, 1, \ldots, T\} \rightarrow Z^+ \cup \{0\}$ specify the flow over time in the network N. More precisely, the number $f(a, t)$ equals the number of flow units entering arc a at time step t. The number of flow units entering arc a at time step t is assumed to be bounded by the capacity of an arc, i.e.,

$$0 \leq f(a, t) \leq c(a) \ \forall a \in A \text{ and } \forall \ t \in \{0, 1, \ldots, T\}. \tag{1}$$

In each time step $t \in \{0, 1, \ldots, T\}$, the flow entering a node $v \in V \backslash \{s, d\}$ has to be at least as large as the flow exiting out of it, i.e.,

$$\sum_{a \in \delta^-(v)} f(a, t - \tau_a) - \sum_{a \in \delta^+(v)} f(a, t) \geq 0 \quad \forall \, v \in V \backslash \{s, d\} \text{ and } \forall \, t \in \{0, 1, \ldots, T\}. \tag{2}$$

Here, $\delta^-(v)$ and $\delta^+(v)$ denote the set of arcs entering and leaving node $v \in V$, respectively. Additionally, it is allowed that flow is held at some node $v \in V$ if $k(v) \neq 0$. To this end, we introduce variables $h(v, T)$ for all $v \in V \backslash \{s\}$ and require

$$0 \leq h(v, T) = \sum_{t=0}^{T} \sum_{a \in \delta^-(v)} f(a, t - \tau_a) - \sum_{t=0}^{T} \sum_{a \in \delta^+(v)} f(a, t) \ \ \forall \, v \in V \backslash \{s\}. \tag{3}$$

The total flow of evacuees leaving source s equals the total flow of the evacuees held at any node $v \in V \backslash \{s\}$ over the time horizon T, i.e.,

$$\sum_{t=0}^{T} \sum_{a \in \delta^+(s)} f(a, t) = \sum_{v \in V \backslash \{s\}} h(v, T). \tag{4}$$

With respect to the constraints from (1) to (4), the *lexicographic maximum dynamic flow problem* asks to send as many flow units from source to sink as possible for each time step $t \in \{0,1,...,T\}$, and as a secondary objective, a maximum number of flow units to nodes other than the sink in the same manner. The latter is subjected to a prioritization of the nodes $v \in V$ from lower to higher priority as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$. This sorting reflects the fact that certain destinations in an evacuation process have different priority. Thus, the objective function of the LexMaxDF problem asks to lexicographically maximize the number of flow units held at the nodes within the pre-specified time horizon T where the nodes are sorted in a given prioritization, i.e.,

$$\text{lex max } \left(h(v_n,T), h(v_{n-1},T), \ldots, h(v_2,T)\right). \tag{5}$$

In the context of evacuation modeling, this objective function can be interpreted as follows. It is $v_n = d$, and thus, a maximum flow from s to d has to be found in the first place. Since $k(d) = +\infty$, the value of this flow is not bounded by the node capacity. Then, let $v_i \neq d$ be the node with highest priority (other than the sink) having positive node capacity $k(v_i)$. Due to the lexicographical optimization, the problem asks for a flow sending as much flow as possible to node v_i among the set of maximal s-v_i flows. This idea is repeated until the flow to the node with lowest priority and positive node capacity is eventually considered.

3 Solution to Lexicographic Maximum Dynamic Flow Problem

Consider a uniform path length (UPL) network $N = (V, A, c(a), k(v), \tau_a, T)$ with prioritized nodes $v_i \in V$ sorted as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$. A directed dynamic network is a uniform path length (UPL) network for which the sum of the transit times on arcs on any possible path from the source s to the node v_i, for all $v_i \in V$, is equal, see Fig. 1. The goal is to solve the lexicographic maximum dynamic flow problem on N in polynomial time using temporally repeated flows. The main idea of the solution procedure of the problem is to find $s - v_i$ paths, for all $v_i \in V : k(v_i) > 0$, at all possible time steps $t \in \{0,1,\ldots,T\}$ with corresponding flow value and send as many units of flow as possible along the paths as long as possible. Such paths can be found by decomposing the flow on solving the *lexicographic minimum cost flow (LexMinCF) problem* on N. The LexMinCF problem asks to lexicographically minimize the cost B_i for sending the number of flow units $\mathbf{f}(v_i)$ at each of the prioritized nodes $v_i \in V$, i.e.,

$$\text{lex min } \left(B_n(\mathbf{f}(v_n)), B_{n-1}(\mathbf{f}(v_{n-1})), \ldots, B_1(\mathbf{f}(v_2))\right) \tag{6}$$

where the transit time $\tau_a \; \forall \, a \in A$ is switched into the cost b_a.

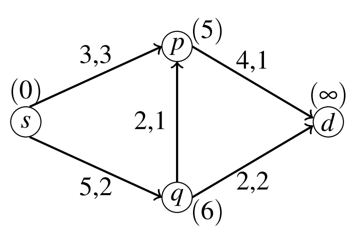


Fig. 1 A uniform path length (UPL) network N with source node s, arc capacity and transit times next to each arc and the node capacity inside the parenthesis near by each node.

The minimum cost flow algorithm in [10], for example, can be applied to solve LexMinCF problem on N repeatedly for each $v_i \in V : k(v_i) > 0$ in given priority order on the corresponding residual network of N with additional arc (v_i, s) with capacity equal to $k(v_i)$ and transit time $-(T+1)$. This yields a set of all $s - v_i$ paths that could be temporally

Proceedings DOI: 10.21467/proceedings.100
ISBN: 978-81-942709-6-6
Series: AIJR Proceedings
ISSN: 2582-3922

repeated from time step zero, denoted as Γ_{v_i}, for each $v_i \in V : k(v_i) > 0$. It is noteworthy to mention that the path γ_{v_i} is a chain of nodes and arcs in the network N starting at the source s and terminating at node v_i.

The limitation of the temporally repeated flow along path on Γ_{v_i} is that it may not induce an optimal solution to the problem or the flow becomes infeasible for some node v_i on N due to fixed node capacity. Thus, it is necessary to find an extended set $\Gamma_{v_i}^E$ that contains all minimum cost $s - v_i$ paths that exist at any time $t \in \{0, 1, ..., T\}$ on the residual network of N with respect to the optimal flow $\mathbf{f}(v_{i+1})$ at previous immediate prioritized node v_{i+1}. It is also necessary to push flow units of corresponding values along each path as long as possible, unless $k(v_i)$ is satisfied. Moreover, the flow is pushed along the paths in $\Gamma_{v_i}^E$ with the strategy of saving unused paths for the use of next less prioritized node v_{i-1} without violating the optimality at v_i. This is assured by selecting the path with highest $F_t(\gamma_{v_i})$, the time step at which the flow along γ_{v_i} stops to get repeated, among the paths $\gamma_{v_i} \in \Gamma_{v_i}^E$with highest $I_t(\gamma_{v_i})$, the time step at which the flow along γ_{v_i} starts to get repeated, at the first and so on. This procedure yields an optimal solution to the LexMaxDF problem on UPL network N in polynomial time.

4 Lexicographic Earliest Arrival Flow Problem

Since it is usually not known when the disaster will actually happen, it is desirable to organize an evacuation in such a way that as many evacuees as possible are saved. An earliest arrival flow problem aims to optimize the evacuation process for every time step within pre-specified time horizon T. A LexMaxDF problem that fulfills the objective function (5) at each time step $t \in \{0, 1, \ldots, T\}$ together with the constraints (1) to (4) is a *lexicographic earliest arrival flow problem*. That is, the objective of a LexEAF problem is to send a maximum number of evacuees at the possible earliest time from the disastrous zone to the safety zone together with relatively safe zones within the given time horizon.

It is clear that every earliest arrival flow is a maximum dynamic flow for given time horizon. However, the converse is not always true for general network. In the following, a solution procedure is proposed that obtains a lexicographic maximum dynamic flow on a typical network and claimed that this flow schedule has an earliest arrival property.

Let us consider the LexMaxDF problem on a uniform path length two terminal series parallel (UPL-TTSP) network $N = (V, A, c(a), k(v), \tau_a, T)$ with prioritized nodes $v_i \in V$ sorted as $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$ with $k(v_i) \in \{0, +\infty\}$. The solution procedure discussed in Section 3 is applied to solve the LexMaxDF problem where the minimum cost flow algorithm [2] is applied to solve the LexMinCF problem. The extended set $\Gamma_{v_i}^E$ induces an optimal dynamic flow for each v_i on N in polynomial time. Moreover, the network N being a two terminal series parallel in structure, this flow has an earliest arrival property [18].

5 Lexicographic Quickest Flow Problem

Let us restrict the node capacity $k(v)$ to be fulfilled as an upper bound as well as a lower bound by the total flow value that is supposed to be held at the node v_i on $N = (V, A, c(a), \tau_a, k(v))$ in the LexMaxDF problem discussed in Section 3. Then the limited node capacity $k(v)$ can be taken as demand, say, $\mu(v)$) at $v : \forall v \in V \setminus \{s\}$. This consideration allows to see a dynamic flow problem on N with demands at nodes and asking for a minimum time to satisfy these demands in given priority order. In the following, we formally define this problem which is termed as *lexicographic quickest flow problem*.

Consider a UPL network $N = (V, A, c(a), \tau_a, \mu(v))$ with prioritized nodes $v_i \in V$ $s = v_1 \preceq v_2 \preceq \ldots \preceq v_n = d$ such that $\sum_i \mu(v_i) = 0$ where $\mu(v_i) \in Z^+ \cup \{0\}$ is the demand at the node v_i. The negative demand at the source s is termed as supply. Moreover, we restrict the arc capacity $c(a)$ for each arc $a \in A$ to be strictly positive. Then the LexQF problem finds a feasible dynamic flow f_{v_i} of given value $\mu(v_i)$ on the network N with prioritized nodes v_i from the source s to the node v_i which sends the given $\mu(v_i)$ units of flow from s to v_i in the minimum number $T(\mu(v_i))$ of time units obeying the capacity constraints (1), the weak flow conservation constraints (2) for time horizon $T(\mu(v_i))$ and the modified form of constraint (3) as

$$\sum_{t=0}^{T(\mu(v))} \sum_{a \in \delta^-(v)} f(a, t-\tau_a) - \sum_{t=0}^{T(\mu(v))} \sum_{a \in \delta^+(v)} f(a,t) = \mu(v) \ \forall v \in V \backslash \{s\} \tag{7}$$

where $T(\mu(v))$ is the minimum time that is required to send $\mu(v)$ units of flow from the source to the node v. Moreover, the Equation (4) holds true due to our consideration $\sum_i \mu(v_i) = 0$. Together with these assumptions, the objective of LexQF problem is

$$\text{lex min } \left(T(\mu(v_n)), T(\mu(v_{n-1})), \ldots, T(\mu(v_2))\right). \tag{8}$$

The existence of lexicographic quickest flow on N follows from the fact that N is a connected network and capacity $c(a)$ is positive for each $a \in A$. The solution procedure to the LexQF problem is similar to the binary search method of solving a quickest flow problem in [4]. Since we are interested in finding such minimum time T_m for each node $v_i \in V : \mu(v_i) > 0$ in a priority order, the maximum flow computation technique developed in Section 3 is adopted as a subroutine of the procedure with necessary modification. Due to the nature of the construction of a maximum flow using this technique, the maximum flow of value $\mu(v_i)$ obtained for time horizon T, could also be possible to find in lesser time horizon T' for some nodes v_i. That is, it cannot be guaranteed that the time T at which the dynamic flow of value $\mu(v_i)$ can be sent to v_i is the minimum time to attain this flow value. One should check whether the same flow value is attained for some lesser time point T'.

6 Solutions with Continuous Time Setting

The lexicographic maximum dynamic flow problem, the lexicographic quickest flow problem and the lexicographic earliest arrival flow problem modeled on the network N with continuous time setting for time horizon T can also be solved efficiently by applying the notion of natural transformation of flows over discrete time setting [6]. The notion states that the amount of flow, say f_d, that arrives at the node w through the arc $a = (v,w)$ at time step t in the discrete time setting is equal to the amount of flow, say f_c, arriving at w through the arc $a = (v,w)$ during the unit interval of time at the beginning of time step t, i.e., $f_d(a,t) := f_c(a,[t,t+1))$ for all $t \in \{0,1,\ldots,T-1\}$.

7 Concluding Remark

The domain of evacuation planning problems based on the network flow model has been flourished with efficient solutions with various network attributes. A common feature of the problems is that the flow function obeys flow conservation constraints at each intermediate node. In particular, maximum dynamic flow problem, earliest arrival flow problem and quickest flow problem have great applicability in evacuation planning problems due to realization of time constraint. In this paper, we studied these problems that lexicogaphically achieve the goals on the network with prioritized intermediate shelters of given capacities. Evacuation problems with intermediate shelters could be extra benefit during disasters. We proposed polynomial time solution techniques for the LexMaxDF problem and LexQF problem modeled on UPL network and for LexEAF problem modeled on UPL-TTSP network. Investigation of solution to these problems modeled on more general network would extend the domain and scope of their applicability in real world evacuation plans.

Acknowledgments: The first author would like to thank University Grants Com-mission, Nepal for partial financial support under PhD Fellowship Award 2016.

References

1. Anderson, E.J., Nash, P., Philpott, A.B.: A class of continuous network flow problems. Mathematics of Operations Research **7**(4), 501–514 (1982)

2. Bein, W.W., Brucker, P., Tamir, A.: Minimum cost flow algorithms for series-parallel networks. Discrete Applied Mathematics **10**(2), 117–124 (1985)
3. Bhandari, P.P., Khadka, S.R., Schafer, L., Ruzika, S.: Lexicographically maximum dynamic flow with vertex capacities. (Submitted) (2019)
4. Burkard, R.E., Dlaska, K., Klinz, B.: The quickest flow problem. Zeitschrift fur Operations Research **37**(1), 31–58 (1993)
5. Dhamala, T.N.: A survey on models and algorithms for discrete evacuation planning network problems. Journal of Industrial & Management Optimization **11**(1), 265–289 (2015)
6. Fleischer, L., Tardos, E.: Efficient continuous time dynamic network flow algorithms. Operations Research Letters **23**(3-5), 71–80 (1998)
7. Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. Operations research **6**(3), 419–33 (1958)
8. Ford, L.R., Fulkerson, D.R.: Flows in networks. Princeton University Press (1962)
9. Gale, D.: Transient flows in networks. Michigan Mathematical Journal **6**, 59–63 (1959)
10. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. Journal of the ACM (JACM) **36**(4), 873–886 (1989)
11. Khadka, S.R., Bhandari, P.P.: Dynamic network contraflow evacuation planning problem with continuous time approach. International Journal of Operations Research **14**(1), 27–34 (2017)
12. Megiddo, N.: Combinatorial optimization with rational objective functions. Math. Oper. Res. **4**, 414—424 (1979)
13. Minieka, E.: Maximal, lexicographic, and dynamic network flows. Operations Research **21**(2), 517–527 (1973)
14. Philpott, A.B.: Algorithms for continuous network flow problems. Doctoral dissertation, University of Cambridge, UK (1982)
15. Philpott, A.B.: Continuous-time flows in networks. Mathematics of Operations Research **15**(4), 640–661 (1990)
16. Pyakurel, U., Dhamala, T.N., Dempe, S.: Efficient continuous contraflow algorithms for evacuation planning problems. Annals of Operations Research **254**, 335–364 (2017)
17. Rebennack, S., Arulselvan, A., Elefteriadou, L., Pardalos, P.M.: Complexity analysis for maximum flow problems with arc reversals. Journal of Combinatorial Optimization **19**(2), 200–216 (2010)
18. Ruzika, S., Sperber, H., Steiner, M.: Earliest arrival flows on series-parallel graphs. Networks **57**(2), 169–173 (2011)
19. Skutella, M.: An introduction to network flows over time. In: Research trends in combinatorial optimization, pp. 451-482. Springer, Heidelberg (2009)
20. Steiner, M.: A survey of earliest arrival flows and a study of the series-parallel case. Diploma dissertation, University of Kaiserslautern (2009)
21. Wilkinson, W.L.: An algorithm for universal maximal dynamic flows in a network. Operations Research **19**(7), 1602–1612 (1971)

MAXIMUM FLOW EVACUATION PLANNING PROBLEM WITH NON-CONSERVATION FLOW CONSTRAINT AT THE INTERMEDIATE NODES

*Phanindra Prasad Bhandari and Shree Ram Khadka
Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal
*Corresponding Author: phanindra.maths@gmail.com

International Conference on Mathematical Optimization (April 8-13,2019, Beijing)

Abstract: In this paper, we formulate the evacuation planning problem with non- conservation flow constraint at the intermediate nodes so that the evacuees can also be sent to relatively safe nodes in between the source and the sink during evacuation. We propose an efficient algorithm for the problem with no transit time and an efficient algorithm which depends on the time horizon for the dynamic case of the problem.

Keywords: *Evacuation Planning Problem, Network Flow, Disaster management.*

1. Problem Formulation

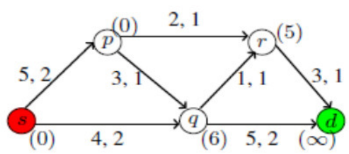


Fig. 1. A network N with the first and the second numbers next to each arc the arc capacity c_e and transit time τ_e , respectively. The number inside the parenthesis nearby each node v represents the node capacity $k(v)$.

For given evacuation network$N = (V, E, c_e, k(v),\ \tau_e, s,\ d, T)$ where V is the set of n nodes and E is the set of arcs,

$$lex\max\big(h(v_n), h(v_{n-1}), \ldots, h(v_1)\big) \quad (1)$$

satisfying the conditions:

$$0\ \le f(e,\theta) \le c(e)\ \forall e\ \in E\ and\ \forall\ \theta\ \in \{0,1,\ldots,T\}. \quad (2)$$

$$\sum_{e\ \in\delta^-(v)} f(e,\theta\ -\tau_e) - \sum_{e\ \in\delta^+(v)} f(e,\theta\ )\ \ \ge 0$$

$$\forall\ v\ \in V\{s,d\}\ and\ \forall\ \theta \in\{0,1,\ldots,T\}. \quad (3)$$

$$0 \le h(v) = \sum_{\theta=0}^{T}\sum_{e\ \in\delta^-(v)} f(e,\theta\ -\tau_e) - \sum_{\theta=0}^{T}\sum_{e\ \in\delta^+(v)} f(e,\theta) \le k(v)$$

$$\forall\ v \in V\ . \quad (4)$$

$$\sum_{\theta=0}^{T}\sum_{e\ \in\delta^+(s)} f(e,\theta) = \sum_{v\in V} h(v)\,. \quad (5)$$

where $f\colon E \times \{0,1,\ldots,T\} \to Z^+$ specify the flow over time horizon T in the network N.

2. Features of the Problem

- Extra evacuees may leave the source who cannot reach the sink.
- Allows to hold evacuees at temporary shelters.
- Helps to provide immediate medication to severely injured evacuees
- Models the situation in which evacuees loss their lives before reaching the safe place.
- Prioritize the evacuees to send to the safe place

3. Solution (Static Case)

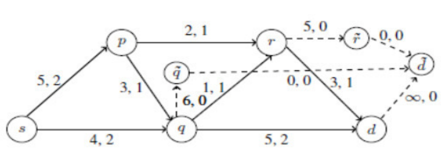


Fig. 2. The transformed network $\widetilde{N}$ of network N depicted in Fig. 1.

Maximum Static Flow Algorithm

1. *Given a static network* $N = (V, E, c_e, k(v),\ s,\ d)$ *with given priority on nodes* $v_i \in V$.
2. *Transform the network* N *into* $\widetilde{N}$.
3. *Find maximum flow from s to* $\tilde{v}_i$ *for all* $i\ =\ 1,2,\ldots,n$ *in given priority order.*
4. *Get lex max static flow on* N.

Optimality and Time Complexity

- Given network N and transformed network $\widetilde{N}$. The maximum flow of evacuees from the source s to node v_i for all $i = 1,2,\ldots,n$ is equal to the maximum flow of evacuees from the source s to $\tilde{v}_i$ where the node $\tilde{v}_i$ is the artificial node corresponding to the node $v_i,\ i\ =\ 1,2,\ldots,n$.
- The optimal flow from the source node s to an intermediate node v_i is independent of the optimal flow at the intermediate node v_j from s, where $c_{v_i}, c_{v_j} > 0;\ v_i,\ v_j\ \in Q; i < j; j \le p$.
- The Algorithm 1 obtains the optimal solution in strongly polynomial of order $O(n^2 m)$, (Goldberg and Tarjan, 1989).

4. Solution (Dynamic Case)

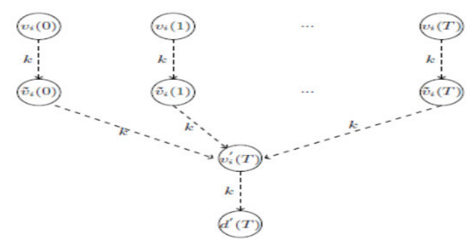


Fig. 3. A part of the time expanded network $\widetilde{N}^T$due to the node capacity k at the intermediate node v_i.

Maximum Dynamic Flow Algorithm

1. *Given a dynamic network* $N = (V, E, c_e, k(v), \tau_e,\ s,\ d, T)$ *with given priority on nodes* $v_i \in V$ *having higher priority to the lower indexed node.*
2. *Transform the network* N *into* $\widetilde{N}$.
3. *Transform the network* $\widetilde{N}$ *into corresponding time-expanded network* $\widetilde{N}^T$ *(Ford and Fulkerson, 1958) with necessary modification.*
4. *Find maximum flow from s to* $\tilde{v}_i$ *for all* $i\ =\ 1,2,\ldots,n$ *in given priority order.*
5. *Get lex max dynamic flow on* N.

Note: Maximum Dynamic Flow Algorithm runs with pseudo-polynomial time complexity.

5. Non-existence of Temporally Repeated Flow

Example 1:

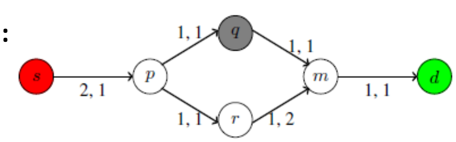


Fig.4. An evacuation network with priority sorting as $q \le d$ and $T = 5$.

Here, the flow value obtained by temporal repetition is not optimal for q since 3 units of flow could be sent at q in total if the path flow along $s - p - r - m - d$ at time $\theta = 0$ is used to send flow at d instead of the path $s - p - q - m - d$ at time $\theta = 1$.

References

1. Ford, L.R. and Fulkerson, D.R. (1958). Constructing Maximal Dynamic Flows from Static Networks. *Operations Research*, 6(3): 419-433.
2. Goldberg, A.V. and Tarjan, R.E. (1989). Finding Minimum-cost Circulations by Cancelling Negative Cycles. *Journal of ACM*, 36: 873-886.

Acknowledgements: The first author is grateful to Silkroad Mathematics Center, China for conference support and University Grants Commission, Nepal for partial financial support to conduct this research work as a PhD Fellowship award.

Certificate of Participation

This is to certify that

Shree Ram Khadka and Phanindra Prasad Bhandari

presented a paper titled

An Efficient Solution Approach to Non-Conservation Maximum Flow Evacuation Planning Problem

at the

11th international conference on Software, Knowledge, Information Management and Application

6 -8 December 2017

Colombo, Sri Lanka

Prof. Nauman Aslam General Co-Chair	Prof. Samantha Thelijjagoda General Co-Chair	Prof. Keshav Dahal General Co-Chair

Technical sponsors IEEE Advancing Technology for Humanity, IEEE Sri Lanka Section

Organized by SLIIT Discover Your Future

INTERNATIONAL CONFERENCE ON APPLIED MATHEMATICS & COMPUTATIONAL SCIENCES

DIT UNIVERSITY
IMAGINE ASPIRE ACHIEVE

Certificate of Participation

Certificate No. 010

This is to Certify that Prof./Dr./~~Mr.~~/Ms. Phanindra Prasad Bhandari

From Tribhuvan University, Nepal ~~Participated~~ / presented an

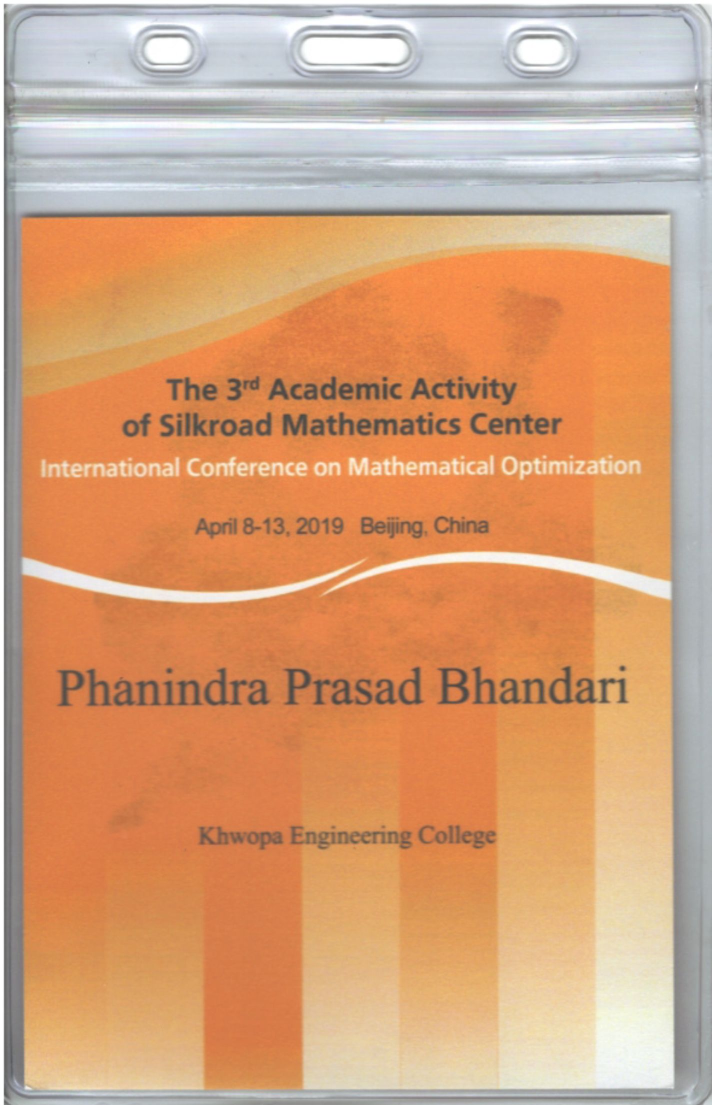
oral paper / poster paper entitled Non-conservation flow constraints Evacuation Planning problems in ICAMCS - 2019, Organized by Department

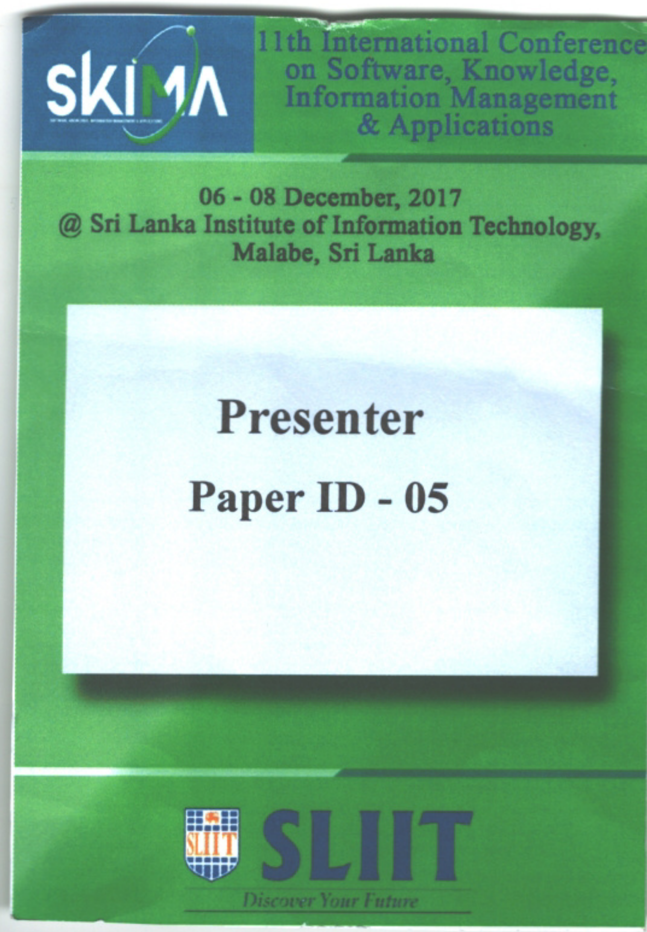
of Mathematics , DIT University, Dehradun & Sponsored by UCOST, Govt. of Uttarakhand from October 17-19, 2019.

ICAMCS 2019

(Dr. Jogendra Kumar)
Convener

(Prof. K.K. Raina)
Vice Chancellor


The 3rd Academic Activity
of Silkroad Mathematics Center
International Conference on Mathematical Optimization
April 8-13, 2019 Beijing, China
Phanindra Prasad Bhandari
Khwopa Engineering College


SKIMA
11th International Conference
on Software, Knowledge,
Information Management
& Applications
06 - 08 December, 2017
@ Sri Lanka Institute of Information Technology,
Malabe, Sri Lanka
Presenter
Paper ID - 05
SLIIT
Discover Your Future


ICAMCS 2019
DIT UNIVERSITY
INTERNATIONAL CONFERENCE
ON
APPLIED MATHEMATICS & COMPUTATIONAL SCIENCES
Oct. 17-19, 2019
PARTICIPANT
Name : Phanindra Bhandari
Affiliation : Tribhuvan University.
Sponsored By


The 12th International Conference on
Numerical Optimization and Numerical Linear Algebra
April 15-18, 2019 Shangrao, Jiangxi
Phanindra Prasad Bhandari
Tribhuvan University, Nepal

"Science for Sustainable World"

7th International Science Conference

Certificate of Appreciation

Presented to

MR. PHANINDRA PD. BHANDARI

In grateful recognition of your valuable participation as

ORAL PRESENTER

in ***7th International Science Conference*** *held at* ***Kathmandu University, Dhulikhel, Nepal*** *on*

10-11 November, 2017 *organized in collaboration with*

World Science Congress.

Prof. Dr. Dulal Chandra Mukherjee
Secretary General
World Science Congress

Prof. Dr. Janardan Lamichane
Chairman, 7th ISC-2017
Kathmandu University

Mr. Viraj Bista
Hon'ble State Minister
Ministry of Science and Technology

APORS 2018

IFORS

11th Triennial Conference of the Association of Asia Pacific Operational Research Societies (APORS)

on

"Operations Research and Development"

August 6-9, 2018

Kathmandu, Nepal

Certificate

This is to certify that

Phanindra Prasad Bhandari

Khwopa Engineering College, Bhaktapur

has presented a paper entitled

"Evacuation Planning Problem Based on Non-Conserving Flow Model"

ORGANIZED BY:
OPERATIONAL RESEARCH SOCIETY OF NEPAL

Prof. Dr. Sunity Shrestha Hada
Chair, APORS 2018
President, ORSN

Dr. Govinda Tamang
General Secretary
ORSN

National Conference on

Mathematics and Its Applications

(NCMA-2017)

January 11-13, 2017, Chitwan, Nepal

CERTIFICATE

This certificate is awarded to

Phanindra Prasad Bhandari

Tribhuvan University

for participating and presenting

a paper entitled

On Non-conserving Flow Aspect of the Evacuation Planning Problem

in the National Conference on

Mathematics and Its Applications

organized by

Nepal Mathematical Society.

Prof. Dr. Tanka Nath Dhamala
President
Nepal Mathematical Society

Prof. Dr. Ishwari Prasad Dhakal
Vice-Chancellor
Agriculture and Forestry University, Nepal

Date: January 13, 2017

"Developing the Society Through Scientific Research & Mathematical Activities"

7th National Conference on Mathematics and Its Applications (NCMA-2019)

Certificate

This certificate is awarded to

Phanindra Prasad Bhandari

Central Department of Mathematics,

Tribhuvan University, Kirtipur

for participating in the

National Conference on Mathematics and Its Applications

organized by

Nepal Mathematical Society.

Prof. Dr. Chet Raj Bhatta
President
Nepal Mathematical Society

Sudarshan Baral
Chief Guest
Minister for Social Development
Province-5, Nepal

January 12-15, 2019
Butwal, Nepal

An International Geoscientific Event

9TH NEPAL GEOLOGICAL CONGRESS (NGC-IX)

Certificate

This certificate is awarded to

Phanindra Prasad Bhandari

for his/her presentation and active participation during **9th Nepal Geological Congress (NGC-IX)** organized by **Nepal Geological Society (NGS)** held in Kathamadu, Nepal on 19-21 November 2018.

Dr. Kabi Raj Paudyal	**Mr. Rajendra Prasad Khanal**	**Dr. Kamala Kant Acharya**
President/Chair	Convener	General Secretary

उत्पादकं यत्प्रवदन्ति बुद्धेरधिष्ठितं सत्पुरूषेण सांख्याः ।
व्यक्तस्य कृत्स्नस्य तदेकवीजमव्यक्तमीशं गणितं च वन्दे ॥

National Conference on
History and Recent Trends of Mathematics

(NCHRTM-2017)

June 2-4, 2017, Kathmandu, Nepal

CERTIFICATE

This certificate is awarded to

Phanindra Prasad Bhandari

Khwopa Engineering College, Bhaktapur, Tribhuvan University

for participating and presenting a paper entitled *Maximum Flow Evacuation Planning Problem without Flow Conservation Aspect* in the Conference organized by Department of Mathematics, Balmeeki Campus, Nepal Sanskrit University in collaboration with Tribhuvan University, Kathmandu University and Nepal Mathematical Society.

Dr. Dinesh Panthi, *Convener*
Associate Professor
Dept. of Math. Balmeeki Campus

Mr. Kishor Gautam, *Chairman*
Head of Department
Dept. of Math. Balmeeki Campus

Dr. Kul Prasad Koirala, *Chief Guest*
Vice-Chancellor
Nepal Sanskrit University

Date: June 4, 2017

TRIBHUVAN UNIVERSITY

CENTRAL DEPARTMENT OF MATHEMATICS
KIRTIPUR, KATHMANDU
NEPAL

Institute of Science and Technology

Certificate of Attendance

Phanindra Bhandari
Central Department of Mathematics, Tribhuvan University, Nepal
has attended the workshop
on
Bilevel Optimization
from 28th February until 7th March of 2017

Trainers
Prof. Dr. Stephan Dempe and Dr. Maria Pilecka
Technical University Bergakademie, Freiberg, Germany

Organizer
Central Department of Mathematics, Tribhuvan University, Nepal

Support
Research Group Linkage Program
Alexander von Humboldt Foundation, Germany

Prof. Dr. Tanka Nath Dhamala
Cooperation Partner
Research Group Linkage Program
Tribhuvan University, Kathmandu, Nepal

Prof. Dr. Stephan Dempe
Principal Resource Person
Faculty of Mathematics and Computer Science
TU Bergakademie, Freiberg, Germany

Dr. Kedar Nath Uprety
Professor & Head

Department of Mathematics and Computer Science
TU Bergakademie Freiberg, Freiberg, Germany

and

Central Department of Mathematics
Institute of Science and Technology, Tribhuvan University, Nepal

in cooperation with

The Alexander von Humboldt (AvH) Foundation

present this

Certificate of Participation

to

Phanindra Bhandari

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

for actively participating in the ***Workshop on Convex Optimization*** *(Optimization Models and Methods for Sustainable Development with focus on Planning, Transportation and Logistics) held at Central Department of Mathematics, Institute of Science and Technology, Tribhuvan University from February 27 to March 07, 2018.*

PROF. DR. STEPHAN DEMPE
(Principal Resource Person and Cooperation Partner)
Department of Mathematics and Computer Science
TU Bergakademie Freiberg, Germany

PROF. DR. TANKA NATH DHAMALA
(Cooperation Partner)
Central Department of Mathematics
IOST, TU, Nepal

Date: March 7, 2018, Kathmandu, Nepal

TU BERGAKADEMIE FREIBERG

Alexander von Humboldt
Stiftung/Foundation

Department of Mathematics
University of Kaiserslautern, Germany

Central Department of Mathematics
Institute of Science and Technology, Tribhuvan University, Nepal

and

Department of Mathematics and Statistics
Mindanao State University - Iligan Institute of Technology, The Philippines

in cooperation with

The German Academic Exchange Service (DAAD)

present this

Certificate of Participation

to

Mr PHANINDRA P BHANDARI

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

for actively participating the Seminar-Workshop on Graph Theory and Optimization with Applications in Industry and Society (Linear, Integer and Multi-Criteria Optimization) held at Central Department of Mathematics, Institute of Science and Technology, Tribhuvan University March12-23, 2018

PROF. DR. FERDINAND P JAMIL
(Cooperation Partner)
Department of Mathematics and Statistics
MSU-IIT, The Philippines

PROF. DR. TANKA NATH DHAMALA
(Cooperation Partner)
Central Department of Mathematics
IOST, TU, Nepal

PROF. DR. SVEN O KRUMKE
(Principal Resource Person)
Department of Mathematics
University of Kaiserslautern, Germany

Date: March 23, 2018, Kathmandu, Nepal

DAAD Deutscher Akademischer Austauschdienst
German Academic Exchange Service

OPT Optimization Research Group

Department of Mathematics
University of Kaiserslautern, Germany

Central Department of Mathematics
Institute of Science and Technology, Tribhuvan University, Nepal

and

Department of Mathematics and Statistics
Mindanao State University - Iligan Institute of Technology, Philippines

in cooperation with

The German Academic Exchange Service (DAAD)

present this

Certificate of Participation

to

Phanindra Prasad Bhandari

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

for actively participating the Seminar-Workshop on Graph Theory and Optimization with Applications in Industry and Society
(Advanced Network Flows: Algorithms and Game Theoretic Views)
held at Central Department of Mathematics, Institute of Science and Technology, Tribhuvan University
March26-April 6, 2017

PROF. DR. SERGIO R. CANOY, JR.
(Cooperation Partner)
Department of Mathematics and Statistics
MSU-IIT, Philippines

PROF. DR. TANKA NATH DHAMALA
(Cooperation Partner)
Central Department of Mathematics
IOST, TU, Nepal

PROF. DR. SVEN O.KRUMKE
(Principal Resource Person)
Department of Mathematics
University of Kaiserslautern, Germany

Date: April 6, 2017, Kathmandu, Nepal

DAAD Deutscher Akademischer Austauschdienst German Academic Exchange Service

OPT Optimization Research Group