EFFICIENT DYNAMIC FLOW ALGORITHMS FOR EVACUATION PLANNING

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

RAM CHANDRA DHUNGANA

AUGUST 2020

EFFICIENT DYNAMIC FLOW ALGORITHMS FOR EVACUATION PLANNING

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

RAM CHANDRA DHUNGANA

AUGUST 2020

DECLARATION

Thesis entitled "**Efficient Dynamic Flow Algorithms for Evacuation Planning**" 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 Prof. Dr. Tanka Nath Dhamala, Central Department of Mathematics, Tribhuvan University.

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.

——————————-

Ram Chandra Dhungana

August, 2020

RECOMMENDATION

This is to recommend that **Ram Chandra Dhungana** has carried out research entitled "**Efficient Dynamic Flow Algorithms for Evacuation Planning**" for the award of Doctor of Philosophy (Ph.D.) in **Mathematics** under my supervision. To my 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.

....

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

August, 2020

TRIBHUVAN UNIVERSITY
CENTRAL DEPARTMENT OF MATHEMATICS
OFFICE OF THE HEAD OF DEPARTMENT
KIRTIPUR, KATHMANDU
NEPAL

***Ref. No.*:**

LETTER OF APPROVAL

Date: 26/08/2020

On the recommendation of **Prof. Dr. Tanka Nath Dhamala**, this Ph. D. thesis submitted by **Ram Chandra Dhungana**, entitled **"Efficient Dynamic Flow Algorithms for Evacuation Planning"** is forwarded by Central Department Research Committee (CDRC) to the Dean, IOST, T.U.

………………………..
Dr. Tanka Nath Dhamala
Professor and Head,
26/08/2020

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

ACKNOWLEDGMENTS

I am very grateful to my supervisor Prof. Dr. Tanka Nath Dhamala who has been a constant source of inspiration and the guiding light for all my endeavors. His trust, suggestions, guidelines, and continuous support made my research work possible with this thesis. I would like to express my sincere thanks to the seniors of our group, Associate professors Dr. Shree Ram Khadka and Dr. Urmila Pyakurel who encourage me to this research. I would also like to acknowledge Dr. Pyakurel for some joint research works.

I extend many thanks to the entire research colleagues in our Optimization Research Group, Nepal for their constructive suggestions and extensive discussions. I am deeply indebted and sincerely grateful to Iswar Mani Adhikari for his support, encouragement, and care in my ups and downs. I would like to express my sincere thanks to all the respected teachers of the Central Department of Mathematics, to the Institute of Science and Technology, and the concerned authorities for their kind co-operation and support throughout my research period. I express my sincere gratitude to the University Grants Commission Nepal and Nepal Mathematical Society for providing me the UGC Ph.D. fellowship and the Nick-Simmons Fellowship, respectively. I am grateful to German Academic Exchange Service for the funding to my research stay at the University of Kaiserslautern, Germany. I am also thankful to Alexander von Humboldt Foundation.

Finally, I express my heartfelt thanks to my father Mahendra Prasad Dhungana, for his unconditional love, sacrifices, and support throughout my life, being my single parents from my early childhood. I feel fortunate to have an understanding and caring better-half Sabita Koirala, who supported me always with a passion for lifting me uphill this phase of life. The dream what I have seen through the eyes of my daughter Shuravi, means a lot for me for my passion. Besides this, I would like to extend my gratitude to my family members and all the well wishers who help me during this work. The completion of this work is also a blessing of my late mother Bhakta Kumari Dhungana.

Ram Chandra Dhungana

August, 2020

ABSTRACT

The large scale calamities caused by different natural or human-created disasters are challenging issues to protect life and their surroundings. A great loss of people and socio-economic damages of our society on such disasters is due to the lack of proper planning and their implementation rather than the disaster itself. These issues draw increasing attention of the researchers towards different aspects of disaster management. It is a complex task to develop a significant and universally accepted solution strategy to handle such issues. During such disasters, the primary concern is to protect the life, property, and their surroundings with a minimum loss as far as possible. There are different solution approaches to have a significant solution for an evacuation planning problem. Contraflow, the lane reversal strategy, is one of the widely accepted solution approaches for evacuation planning as it maximizes the outbound capacities of roads by reversing the required road directions and makes the traffic smooth. This significantly increases the flow value and decreases the evacuation time.

The abstract flow model deals with the flow paths (routes) that satisfies the switching property. This concept can be embedded in the contraflow technique to have the mathematical formulation on abstract contraflow models with efficient algorithms for solving such abstract contraflow problems. In this work, different efficient solution procedures are presented for maximum dynamic, lexicographically maximum, and earliest arrival abstract contraflow problems. This approach maximizes the flow value in a given time and seeks to eliminate the crossing conflicts. The earliest arrival flow problem is one of the most important aspects of evacuation planning with a given capacity and travel time. The objective of the problem is to send the maximum number of evacuees from the given sources to the sinks as quickly as possible. It maximizes the flow value at each time instances simultaneously. Here, we study the earliest arrival flow problem with the contraflow approach having supplies and demands in abstract network.

During the evacuation planning problem, one of the essential components is the facility location as it correlates the pre- and post-disaster management. Appropriate facility locations and transportation facilities play a vital role in the solution of evacuation planning problems. Here, the network facility location and the contraflow approach are incorporated into the flow models and some efficient algorithms are presented to locate the facility with an objective of minimum flow loss on the evacuation network. Our facility location contraflow solutions obtain optimal plans concerning the given and as well as arbitrary locations.

With limited resources, it is not an easy task to develop a universally accepted model to handle different aspects and challenges of the evacuation planning problem. However, the budget-constrained network flow improvement approach plays a significant role to evacuate the maximum number of people within the given time horizon for the budget provided. We consider an evacuation planning problem that aims to shift the maximum number of evacuees from a danger area to a safe zone in limited time under the budget constraints for network modification. In this work, different flow improvement strategies for fixed switching costs will be investigated namely integral, rational, and either to increase the full capacity of an arc or not at all. A solution technique on a static network is extended to the dynamic one. Moreover, we introduce the static and dynamic maximum flow problems with lane reversal strategy and also propose efficient algorithms for their solutions. Here, the contraflow approach reverses the direction of arcs concerning the lane reversal costs to increase the flow value. As an implementation of an evacuation plan may demand a large cost, the solutions proposed in this thesis with budget-constrained problems play an important role in practice.

In this thesis, the contraflow models and their solutions strategies have been established and investigated in an abstract network topology. To allocate the facility during the evacuation process FlowLoc problems and their solution have been introduced in the evacuation network. The arc switching costs have been considered for the first time in the evacuation network. These optimization methods play significant roles in maximizing the flow and minimizing the evacuation time, and also have the great support for logistics and emergency vehicle movements in disasters.

LIST OF ACRONYMS AND ABBREVIATIONS

AAN	auxiliary abstract network 50
ATEN	abstract time-expanded network 47
BEP	bus-based evacuation planning 42
DFP	dynamic flow problem 19
EAACF	earliest arrival abstract contraflow 61
EAAF	earliest arrival abstract flow 61
EACFP	earliest arrival contraflow problem 40
EATCFP	earliest arrival transshipment contraflow problem 40
FPTAS	fully polynomial-time approximation schemes . . 11
LMACF	lexicographically maximum abstract contraflow 59
LMAF	lexicographically maximum abstract flow . . . 48
LMDCFP	lexicographically maximum dynamic contraflow problem 39
MACF	maximum abstract contraflow 50
MAF	maximum abstract flow 48
MCFF	minimum cost fixed flow 98
MCFP	minimum cost flow problem 19
MDACF	maximum dynamic abstract contraflow 54
MDCFP	maximum dynamic contraflow problem 39

LIST OF SYMBOLS

$G = (V, A)$	directed graph	11
$H = (V', A')$	sub-graph	13
$I(\tau)$	capacity improvement function over time	94
$I_T^{e(\tau)}$	capacity improvement function in time-expanded network	95
I	capacity improvement function	91
$N = (E, \Gamma, u, \theta, T, S, D)$	multi-terminal abstract dynamic network	46
$N = (V, A, u, \theta, T)$	dynamic network	15
$N_T = (V_T, A_T, u_T)$	time-expanded network	28
T	egress time	14
U_e	upper bound to increase the capacity of e	91
$U_{\max}$	maximum of upper bounds	93
V^+	set of sources	14
V^-	set of sinks	14
V^0	set of intermediate vertices	14
V_T	set of vertices in time-expanded network	28
V	set of vertices	11
Γ_e	subset of paths incident to e	16
Γ_T^σ	set of all temporal paths γ^σ bounded by T	47
Γ_e^τ	set of path incident to e at τ	16
Γ	set of paths	15
γ^σ	temporal paths with waiting time	47
$\lceil \cdot \rceil$	ceiling function	99
$\mathbb{R}$	set of real numbers	14
$\mathbf{T}_c = \{[0, 1), \ldots, [T, T + 1)\}$	partition of T in continuous time model	14
$\mathcal{T} = \{0, 1, \ldots, T\}$	discretization of egress time T	14
first(γ)	first element of path γ	48
last(γ)	last element of path γ	48
ω	weight function defined on path	46
$\overleftarrow{\gamma}$	backward path corresonding to $\overrightarrow{\gamma}$	21
$\overleftarrow{e} = (w, v)$	backward arc corresonding to $\overrightarrow{e}(v, w)$	21

$\overleftrightarrow{A}$	set of arcs in residual network 22
$\overleftrightarrow{N} = (V, \overleftrightarrow{A}, u_f, s, t)$	two-terminal residual network 22
$\overleftrightarrow{\Gamma}_e^\tau$	collection of paths incident to e at τ in residual network 16
$\overleftrightarrow{\Gamma}_{\overleftarrow{e}}^\tau$	collection of path incident to reverse of e at τ in residual network 16
$\overrightarrow{\gamma}$	forward path corresonding to $\overleftarrow{\gamma}$ 21
$\overrightarrow{e} = (v, w)$	forward arc corresonding to $\overleftarrow{e}(w, v)$ 21
σ	waiting time function 47
θ_γ	transit time of path γ 16
θ	transit time function 14
$\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, s, t)$	two-terminal abstract auxiliary network 49
$\tilde{A}$	set of arcs in auxiliary network 21
$\tilde{N} = (V, \tilde{A}, \tilde{u}, \theta, T)$	dynamic auxiliary network 21
$\tilde{U}_{\max}$	maximum of upper bounds in $\tilde{N}$ 98
$\tilde{\Gamma}$	set of paths in auxiliary network 21
$\tilde{\theta}$	transit time function in auxiliary network 21
$\tilde{u}$	capacity function in auxiliary network 21
$b(\tau)$	capacity incremental cost over time 94
$b_T^{e(\tau)}$	capacity incremental cost in the time-expanded network 95
b_e	capacity incremental cost of e 91
c	cost function 14
d^+	supply function defined on element 48
d^-	demand function defined on element 48
d	demand function 14
$f_T^{e(\tau)}$	flow function in time-expanded network 29
$f_e(\tau)$	flow entering on arc e at time τ 15
f	static flow function 15
$g^1 \geq_L g^2$	g^1 is lexicogaphically bigger than g^2 48
g^c	flow function over continuous time on path . 16

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

We are living under the great threat of different natural and human-created disasters, such as earthquakes, volcanic eruptions, landslides, hurricanes, floods, tsunamis, wildfires, terrorist attacks, and chemical spills. Most of the disasters are uncertain and are unavoidable. The choice of an optimal shelter location and the necessary support to their humanitarian logistics are equally demanding and most challenging issues during such emergency situations. Nevertheless, one must take into consideration traffic in the evacuation network as the duration of the evacuation process is prominently influenced by the traffic. Due to the importance of the best possible choice of shelter locations with minimum flow loss or minimum increase in the network clearance time, efficient evacuation planning has drawn the interest of current researchers.

The development of efficient models and algorithms with a universally accepted solution approach to address different issues and demands on the evacuation planning problems is always challenging. To find efficient transportation routes during the evacuation, the prominent dynamic network flow models have been widely used to get the efficient evacuation routes that were first introduced and investigated in [1]. There exist different models and algorithms for buildings, stadiums, ships, districts, cities, or whole sub-national region evacuation based on their scenarios. For different variants of the dynamic flow problems and their corresponding results, we refer to [2, 3, 4, 5, 6, 7, 8, 9].

During the evacuation management, most of traffic delays occur on roads due to different facility locations around the roads. Shifting people with efficient routes to the sinks with proper facility locations is an innermost challenge to manage a regional evacuation plan. An evacuation optimizer seeks for such a plan in the evacuation network which is capable of the efficient transfer of a maximum number of evacuees from the sources to the sinks as quickly as possible. There are various approaches for the mathematical formulations in the evacuation planning problems like traffic simulation, fluid dynamics, control theory, variational inequalities, network flow formulations, etc. Among them, the formulation based on the network flow methodologies is the most efficient, [4, 5, 10].

The network flow problems are to find maximum flow or minimum cost flow in a given network. The maximum dynamic flow problem in two-terminal networks is solved polynomially using a static minimum cost flow solution, [11]. A flow maximization seeks to send as much flow amount as possible within a time-bound. A large number of researchers have studied different flow models for various objectives such as the earliest arrival flow to maximize the flow at every possible time; quickest flow to shift given amount of flow in minimum time; lexicographically maximum flow to maximize the flow in given priority order and quickest transshipment problem to satisfy given demand and supply in minimum time. These dynamic flow models have widely been used in solving several evacuation planning problems. Static flow solutions are the building blocks for dynamic flow solutions. Usually, the evacuation plans should respect given time-bound that may be continuous or discrete. Authors in [12] show that approximate continuous time solution can be obtained by applying the natural transformation to a discretized time solution. Most of these models, except that of cost minimization problems themselves, consider the travel time on an arc as only the cost on it and do not take care of any additional costs that occurred during the evacuation plans. For details, we refer to [4], [5], and the references therein.

The contraflow approach is another emerging and widely accepted model for evacuation planning. It increases the outbound road capacities by reversing the direction of roads towards safe destinations. Kim et al. [13] give first integer programming formulation and presented different heuristic solutions for large scale evacuations. They showed that the problem of minimizing the evacuation time is $\mathcal{NP}$-hard. For a two-terminal abstract contraflow network with the symmetric minimum dynamic cut capacity of each element, the contraflow configuration improves the 100 percent of flow value, [14].

The maximum dynamic contraflow problem has been introduced in [15, 16] and solved with a polynomial algorithm for a two-terminal general network. They proved that the problem is $\mathcal{NP}$-hard for a multi-terminal network. Authors in [17, 18, 19, 20, 21] introduced the earliest arrival, quickest transshipment, and lex-maximum dynamic contraflow problems in both discrete and continuous-time settings. The former is solved in strongly polynomial-time in a two-terminal network and with pseudo polynomial-time in a general network. They presented a polynomial-time algorithm for the quickest transshipment contraflow problem. For the given priority ordering, the lex-maximum dynamic contraflow problem is solved in polynomial-time for a multi-terminal network. For the fixed supply and demands, the earliest arrival transshipment contraflow has been introduced in [21, 22] in discrete and continuous times, respectively. They solved the problem in multi-source and single sink networks as well as a single source and multi-sink networks in polynomial-time complexity. Moreover, they presented approximation algorithms to compute the approximate earliest arrival transshipment contraflow for multi-terminal networks, [14, 22]. The contraflow approach is generalized on a lossy network in [23]. They solved the generalized maximum dynamic and generalized earliest arrival contraflow problems on a two-terminal lossy network. For more details, we refer to [24].

A class of contraflow algorithms including a variety of computational experiments are developed in [25]. The technique of lane reversals is beneficial for various purposes, for example crossing eliminations, logistic supports, and use of emergency vehicles and facility locations. The contraflow approach with crossing elimination, facility location-allocation, and partial lane reversal strategies are introduced in [26, 27, 28], respectively. The partial lane reversal approach makes use of non-reversed arcs for supporting facilities and emergency logistics. Considering the influence of intersections, an improved critical-road model has been investigated to find the optimal contraflow links, [29].

In this work, we introduce and investigate abstract contraflow models by combining the concepts of abstract flow and contraflow approaches. We present some efficient solution procedures to maximize the achievable number of evacuees at a minimum time by reducing the possible crossing conflicts, [26, 30]. Our contributions resemble the results in contraflow on non-abstract networks presented by different authors [19, 20, 16]. We reverse the paths to increase the path capacities whenever necessary by applying the concepts of [14, 16]. Then the problems are solved by using abstract flow algorithms of [31, 32].

The set of locations can be fixed in advance or it can be arbitrary. The former approach has its advantage in allocating facilities at predefined locations that a policymaker requires, however, the latter one gives us the flexibility of allocating a facility among the arbitrary locations. Based on given possible locations, we introduce the two-terminal FlowLoc problem in a dynamic network and propose an efficient algorithm for its solution. The corresponding results for two-terminal FlowLoc static and dynamic contraflow problems are presented for both of the cases, [27, 33].

For provided limited resources, it is not possible to select all arc reversals as demanded by the optimal contraflows. We investigate contraflow problems with fixed budget constraints distributed to the arc reversals. The total given budget allows us to reverse only a certain percentage of arcs in a given network. We introduce the maximum dynamic flow improvement problem and also the maximum contraflow improvement problems in both static and dynamic networks. Then we propose polynomial-time algorithms to solve these problems. To the best of our knowledge, this is the first attempt to incorporate the issues of arc reversal costs on contraflow problems subject to the given total budget constraint. As arc reversals require a lot of costs during the emergency period, this approach is more practicable in implementing the contraflow algorithms, [34, 35]. The network flow models and the solutions on static and dynamic flow improvement problems are presented. The contraflow models and their solution procedures with unit switching costs on arcs for the flow improvements are also established.

The structure of this thesis is as follows. In Chapter 2, the basic denotations, mathematical models, compact literature review on network problems, contraflow problems, and some of the applications of network flow problems are highlighted. Chapter 3 presents abstract flow problems with different objectives and introduces maximum static, maximum dynamic, lexicographically maximum, earliest arrival abstract contraflow problems with their solutions by integrating an abstract flow and contraflow approaches. Network facility location (FlowLoc) problem and contraflow approach have been integrated to introduce ContraFlowLoc problems in Chapter 4. The budget constraint flow improvement problem in a static network has been investigated, the dynamic budget constraints flow improvement has been introduced and solved in Chapter 5. Moreover, the maximum contraflow problems with budget-constrained switching costs have been presented with their algorithms. Chapter 6 concludes the dissertation by opening further direction of research.

1.2 Rationale

An evacuation network is interpreted by a directed graph where the intersections of roads are represented by nodes, road segments between nodes are represented by arcs, and routes between two nodes are taken as directed paths. The places where evacuees are gathered and start to move at risk are considered as source nodes (danger zones) and the safe destinations where they are supposed to arrive are sink nodes (safe zones). Each node has a non-negative integer capacity which bounds the maximum possible flow amount through it. Every arc has a cost or a transit time assigned to it. Flow in the network is considered as the evacuees or the vehicles carrying evacuees.

The abstract flows introduced in [36] generalizes the concept of paths by replacing the underlying network configuration. The maximum dynamic abstract flow problem and its solution procedure have been investigated, [32]. The lexicographically maximum abstract flow problem has been investigated in [31]. This approach makes the use of the switching property which seeks to reduce the crossings at intersections. Some of the important lane-based routing strategies for reducing the delays that significantly reduce and eliminate crossing and merging conflicts at intersections have been studied in [37]. In a lane-based routing plan, selecting and turning options at intersections are restricted to improve traffic flow away from a hazardous area. Intersections with potentially significant delays can be temporarily transformed into an uninterrupted flow facility which is the most beneficial aspect of this routing.

The location theory was introduced in [38] with applications for industries. Different discrete network location models and algorithms have been investigated in [39]. For details, we refer to [40]. The influence of facilities on the walking speed and the walking behavior of pedestrians have been investigated so far. However, [41] uses location theory to improve the existing evacuation models where two different models, network flow and location theory have been integrated to introduce FlowLoc theory in the evacuation modeling.

Different types of network modification problems exist in the literature. Generally, the original network is assumed to be not modifiable in the sense that capabilities or costs remain fixed as in the given network. However, this assumption is not valid in many of the real evacuation scenarios. For example, the capacity of an arc can also be increased up to some limit subject to some capacity incremental cost. For this, a fixed budget can be distributed to increase capacities in the network such that the network topology is modified and an objective, for instance, the flow, with respect to new capacities is maximized. There are three variants of this improvement strategy that deal with rational, integral, and either of the all possible or not at all capacity values in [42]. The first two variants are polynomial-time solvable, while the last one is $\mathcal{NP}$-hard even in the cases of bipartite and series-parallel graphs. This third variant called the $0/1$ maximum flow improvement strategy is equivalent to the maximum flow fixed cost problem which is a bi-criteria optimization problem where the flow has to be maximized under the budget constraints.

1.3 Objectives

The development of efficient models, algorithms, and solution approaches with universally accepted strategies to address different issues and demands on the wide varieties of network topologies and objectives are always challenging in evacuation planning problems. In this study, our focus is on the followings:

- To solve the universally quickest transshipment problem in general evacuation network that minimizes the time when the last supply reaches the sink and maximizes the amount of supplies which have reached the sink at every time efficiently.
- To introduce and investigate the abstract contraflow models by combining the concepts of abstract flow and contraflow approaches to develop the ContraFlowLoc model to locate the facilities in the given network to maximize the flow in an updated network.
- For limited resources, it is not possible to select all arc reversals as denoted by the optimal contraflows. For such situations, we seek to develop the contraflow problems with fixed budget constraints to introduce the maximum dynamic flow improvement problem and the maximum contraflow improvements in different network structures.

CHAPTER 2

PRELIMINARIES

This chapter presents some of the basic terms and their denotations, definitions, and models that are frequently used in the thesis. This helps for the consistency of their notations and discussion on its literature. All these considered topics are the basic foundations in combinatorial optimization, discrete mathematics, integer programming, graph theory, and network optimization. For more details on the topics, we refer, [4, 43, 44, 45, 46].

2.1 Basic Denotation

An optimization problem is a set $\langle \mathcal{I} \rangle$ of all instances (Ω, c) of a given problem, where Ω represents the set of all feasible solution and $c : \Omega \to \mathbb{R}$ is the cost function. If all the variables are continuous/discrete, the problem can be considered as continuous/discrete optimization problem. Some systematic operations are to be performed on the given instance $\mathcal{I}$, as well as the decision for the next step should be made before extracting the optimal solution. Such a systematic list of necessary operations including the decision for the consecutive steps is said to be an algorithm.

It is considered that all the simple operations need a one-time unit, and the total number of such time units is the running time of the algorithm which is also known its time complexity. The number of values of input data that should be stored during the execution of an algorithm is said to be the space complexity of the algorithm. It is measured as the necessary computer memory used. The reduction in both of such complexities of an algorithm is one of the special interest of an optimizer. The uniform cost deals with both sorts of complexities as the cost which extremely relay on implementation, coding language, and the machine where the algorithm is performed. The random access machines count the uniform costs of the algorithm. Large numbers may demand a longer computational time as compared to the small numbers, such costs are said to be logarithmic costs .

The input of an instance affects complexity. Let $n := |\langle\mathcal{I}\rangle|$ be the number of bits required to an instance $\langle\mathcal{I}\rangle$ which is also known as the size of the instance. The time complexity of the algorithm inclines on the input is bounded by $\alpha(n)$, where $\alpha(n) : \mathbb{N} \to \mathbb{N}$ is the runtime function. The worst-case running time should be considered as the instances and their attributes are not known in advance. Let $\mathcal{I}_c$ be a collection of valid instances for a problem then the worst-case running time is defined by $\max\{\min\{\alpha(\langle\mathcal{I}\rangle) \mid \alpha$ runtime function for $\langle\mathcal{I}\rangle\} \mid \langle I\rangle \in \mathcal{I}_c\}$.

If the complexity $\alpha(n)$ is bounded by a polynomial in n, then it is said to be polynomial-time complexity and the algorithm is said to be polynomial-time algorithm. If the time complexity is not polynomial, then it is said to be super-polynomial-time complexity or exponential-time complexity. Similarly, if the complexity $\alpha(n)$ is bounded by a linear function of n then it is said to be linear-time complexity. The interest of the optimizer is to find the bound of the complexity rather than finding the exact running time of the algorithm. The bound of the running time is measured by $\mathcal{O}-$notation. Suppose $\beta : \mathbb{N} \to \mathbb{N}$, then $\mathcal{O}(\beta) := \{\alpha : \mathbb{N} \to \mathbb{N} \mid \exists\ c \in R^+ : \exists\ n_o \in \mathbb{N} : \forall n \geq n_o : \alpha(n) \leq c\beta(n)\}$, is a collection of functions whose order can not be greater than the order of β. The logarithmic term may not affect the order so that it will be omitted from the expression. Hence, $\tilde{\mathcal{O}}(\beta) := \{\alpha : \mathbb{N} \to \mathbb{N} \mid \alpha \in \mathcal{O}(\beta(n). \log_k(\beta(n)))\}$. A polynomial-time algorithm is said to be strongly-polynomial if its complexity is only on the size of the input (without logarithmic term), otherwise it is known as weakly-polynomial . If the time complexity of an algorithm is polynomial, then it is said to be an efficient algorithm. A problem is said to be a number problem if κ cannot be bounded by $n := |\langle \mathcal{I} \rangle|$, where κ is the maximum value in the input instance $\langle \mathcal{I} \rangle$. If the running time of any number problem is bounded by a polynomial in both $|\langle \mathcal{I} \rangle|$ and κ, then that is with pseudo polynomial-time complexity. The complexity of such a pseudo polynomial algorithm heavily depends on the size of numbers of the instance.

Any problem having either yes or no as its solution is known as the decision problem. Such instances with yes and no solutions are said to be yes-instance and no-instance, respectively. Thus, the decision problem ensures the solution of the optimal problem is possible or not. The set of optimization problems having a polynomial-time deterministic algorithm is in class $\mathcal{P}$ where the class $\mathcal{NP}$ consists of all those decision problems whose yes solutions can be checked in polynomial-time but they do not have any polynomial-time deterministic algorithms ($\mathcal{NP}$ class problems can be solved in polynomial-time using non-deterministic algorithms). This implies that, $\mathcal{P} \subseteq \mathcal{NP}$. Two problems A and B are said to be reducible problems, if an algorithm transforms an instance of B into an instance of A in polynomial-time such that any algorithm for A accepts $\langle I_A \rangle$ if and only if $\langle I_B \rangle$ is with a yes-instance for B.

A problem B is said to be an $\mathcal{NP}$-hard if each problem in $\mathcal{NP}$ can be polynomially reducible to B for all $A \in \mathcal{NP}$. In addition, if $B \in \mathcal{NP}$ then B becomes $\mathcal{NP}$-complete. Hence, $\mathcal{P} = \mathcal{NP}$ is possible only when a polynomial algorithm for any $\mathcal{NP}$-hard problem is possible. But, the results are presented assuming that $\mathcal{P} \neq \mathcal{NP}$. An $\mathcal{NP}$-hard problem is said to be weakly $\mathcal{NP}$-hard if it has a pseudo polynomial algorithm. If all numbers in the input of the $\mathcal{NP}$-hard problem are bounded by $|\langle \mathcal{I} \rangle|$, then such a problem is said to be a strongly $\mathcal{NP}$-hard. Developing a polynomial-time algorithm for an $\mathcal{NP}$-hard problem is almost impossible until $\mathcal{P} \neq \mathcal{NP}$ remains true. Hence, many algorithms have been developed to approximate the solution. Williamson and Shmoys [47] have provided many ideas to design approximation algorithms.

Any polynomial-time approximation algorithm that gives a solution whose corresponding value can be written as multiple of κ for $\kappa > 0$ with the optimal value be OPT is known as κ-approximation algorithm. Such an approximation algorithm provides the solutions with the corresponding values at least OPT/κ and at most $\kappa \cdot OPT$ for maximization and minimization problems, respectively. The sufficiently small κ for the algorithm will minimize the gap between optimal and approximate solutions. For any $\epsilon > 0$, $(1+\epsilon)$-approximation algorithm can be obtained by a polynomial-time approximation scheme (PTAS) which is a family of algorithms AL_ϵ. The efficient $(1+\epsilon)$-approximation can be obtained by the fully polynomial-time approximation schemes (FPTAS), which is a family of approximate algorithms having polynomial-time complexity, depending in the input size and in $1/\epsilon$.

The performance of the decision algorithm can be improved by limiting the number of calls of the algorithm without disturbing the feasibility of the problem. But the issue of this approach is to find minimal parameter satisfying the condition. Let $\aleph$ be the parameter. The binary search could be applied to fix the parameter if the minimum and the maximum values are known, which demands $\mathcal{O}(\log(\lambda_{\max} - \lambda_{\min}))$ calls of the algorithm as its subroutine. But it can not ensure strongly polynomial complexity. The parametric search presented by Megiddo [48] improves this procedure in strongly polynomial-time. For this each variable of the algorithm should be replaced by linear functions as a results only of additions, scalar multiplications and comparisons will be in the algorithm. This transformation could be completed in linear time so that it will not affect the complexity.

2.2 Network

A network is considered as a graph with different attributes such as capacities, transit times, loss/gain factors, and costs. The networks are also considered as graph and vice versa in different literature. Mostly, a network is taken as a directed graph with various attributes whereas a graph can be either directed or undirected, and may be without any attributes. A directed graph $G = (V, A)$ consists of a set of vertices V and a set of arc $A \subseteq V \times V$. Various attributes such as waiting time, capacity, cost, supply, and demand can also be considered at vertices. In this thesis mainly the demands and supplies are considered as the attributes. Here, the terms graph and network will be assumed interchangeably and will be mentioned whether the graphs are directed or not. Assumed attributes will be explained whenever necessary. A sketch of the network is shown in Figure 1. The network with time components is said to be dynamic network, otherwise, it is said to be static. Figure 1 is the dynamic network as it considers transit time.

An evacuation network is interpreted by a directed graph where the intersections of roads are represented by nodes, road segments between nodes are represented by arcs, and routes between two nodes are taken as directed paths. The places where evacuees are gathered and start to move at risk are considered as source nodes (danger zones) and the destinations where they are supposed to arrive are sink nodes (safe zones). Each node has a non-negative integer capacity which bounds the maximum possible flow amount through it. Every arc has a cost or a transit time assigned to it. Flow in the network is considered as the evacuees or the vehicles carrying evacuees.

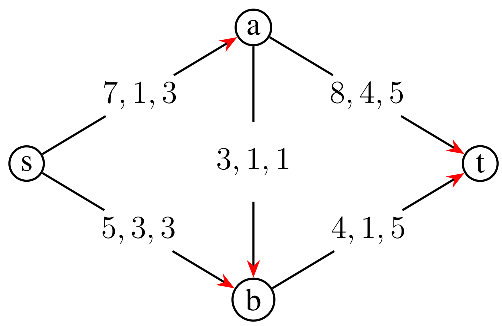


Figure 1: Network with capacities, transit time, and costs, respectively.

An arc $e \in A$ is a pair of vertices $(v, w); v, w \in V$, here v is the tail and w is the head of the arc. If the head and tail of the arc are identical then it is said to be a loop ((v, v) is a loop). An arc $e' = (v, w)$ is the parallel arc of the arc e if $e \neq e'$ whereas the arc $e^r = (w, v)$ is the reverse arc of $e = (v, w)$. The numbers of vertices and arcs are denoted and defined by $n := |V|$ and $m := |A|$, respectively. We neither consider loop nor parallel arcs. But, the loops and parallel arcs can be split up by taking the help of additional vertices. Thus, arcs will be identified by their tail and head vertex.

Let s and t be two distinct vertices. A series-parallel network can be defined by the following recursive definition. An arc (s, t) is a series-parallel network. Let H_1 and H_2 be two such networks with vertices s_1, t_1 and s_2, t_2, respectively. The series-composition defines a series-parallel network H' for $t_1 = s_2$. Likewise, the parallel-composition defines a series-parallel network H' for $s_1 = s_2$ and $t_1 = t_2$. Such a series-parallel network can be used to give better algorithms, to prove the hardness of the problem even in the simpler network topology.

A collection of arcs $C := (X, \overline{X}) \subseteq A$ is said to be an $s - t$-cut if it holds: $X \cap \overline{X} = \emptyset,\ X \cup \overline{X} = V,\ s \in X$ and $t \in \overline{X}$. If c_e is the cost function on arcs then the value of $s - t$-cut is defined by

$$\sum_{e \in (X, \overline{X})} c_e.$$

Suppose A_v^+ be the set of arcs leaving a vertex v, taken as outgoing arcs and A_v^- be the set of arcs entering the vertex, considered as the set of incoming arcs, and A_v as the sum of the two:

$$\begin{aligned} A_v^+ &:= \{(v, w) \in A \mid \text{for some } w \in V\}, \\ A_v^- &:= \{(w, v) \in A \mid \text{for some } w \in V\}, \\ A_v &:= A_v^+ \cup A_v^-, \text{for } v \in V. \end{aligned}$$

Here, the arcs $e \in A_v$ are incident to v whereas $v, w \in V$ are adjacent vertex of $e = (v, w)$. An undirected graph consists the set of vertices and edges where an edge is a set $\{v, w\}$ containing two vertices $v, w \in V$. An undirected graph could be constructed from directed graph by forgetting the orientation of arcs. A directed graph $H = (V', A')$ is said to be sub-graph of $G = (V, A)$ if $V' \subseteq V$ and $A' \subseteq A$. A sub-graph is said to be simple if it consists a single connected sequence of arcs. A sequence of arcs $(e_1, e_2, \ldots, e_k)$ is said to be $v - w$-sequence where $e_i := (v_i, v_{i+1}) \in A$, $v_i = v$ and $v_{k+1} = w$. The length of the sequence is the total number of arcs belonging to the sequence. A sequence is said to be path if $v_i \neq v_j$ for all $i \neq j$. If the starting and ending vertices are the same in the sequence, then it is said to be a cycle. A $v - w$-sequence S of arcs or edges can always be decomposed into a $v - w$-path γ and zero or more cycles. The complexity of such decomposition is $\mathcal{O}(|S|)$.

2.2.1 Attributes

Let $u : A \times \mathcal{T} \to R_{\geq 0}$ be a capacity function which limits the flow rate that can enter on an arc at any point of time. Capacities limit the total amount of flow on an arc if it is static network flow. The transit time function $\theta : A \to \mathbb{R}_{\geq 0}$ ensures the amount of time requiring to traverse a unit of flow on an arc, *i.e.*, if a unit of flow starts to travel the arc $e = (v, w)$ at time t, arrives at w at time $t + \theta_e$. A constant $T \in R_{\geq 0}$ is the egress time (time horizon) specified in dynamic network flow problem to fix the deadline of flow, *i.e,* there will not be any flow in network after this time. The egress time T is denoted by $\mathcal{T} = \{0, 1, \dots, T\}$ and $\mathbf{T}_c = \{[0, 1), \dots, [T, T+1)\}$ in discrete and continuous time models, respectively.

The price that should be accepted for sending a unit of flow though an arc is called cost. The cost function is defined as $c : A \to R$. The cost $c_e f_e$ is the total costs required for sending f_e units of flow through the arc e. Our focus is on linear costs. Distance between two can also be considered as a cost. The minimum distance from the vertex v to w is denoted by $dis(v, w)$ and the path from v to w is known as the shortest path between the these vertices. Dijkstra [49] provided an algorithm for non-negative arc lengths. On the other-hand, the shortest path can be extracted by using Moore-Bellman-Ford-Algorithm [50, 51, 52] if there is some arcs with negative lengths but not any negative length cycles.

Suppose $\lambda : A \longrightarrow \mathbb{R}_{\geq 0}$ be a gain function which ensures the changes in the flow while traversing an arc. For example, evaporation, interest, taxes can be modeled by using such a gain function. The $\lambda_e f_e$ denotes the resulting flow of f_e after traversing the arc. The network with $\lambda_e < 1$ for all $e \in A$ is said to be lossy network. If $\lambda = 1$ for all $e \in A$ then the network becomes the classical one. At vertices, supplies and demands are considered as attributes. Let $d : A \to \mathbb{R}$ be supply-demand function with supply $d_v > 0$ and demand $d_v < 0$ for $v \in V$. Vertices $v \in V$ with $d_v > 0$ are said to be sources and those d_v are said to be supplies. Similarly, vertices $v \in V$ with $d_v < 0$ are said to be sinks and d_v are said to be demands. The sources and sinks are also known as terminals . All other vertices are said to be intermediate vertices. For convention, set of sources, sinks and intermediate vertices are represented by V^+, V^-, and V^0, respectively. If $V^+ = \{s\}$ and $V^- = \{t\}$ then the network is called a two-terminal network, otherwise it becomes a multi-terminal network . If the demands and supplies are finite then the respective sum must be equal, *i.e.*, $\sum_{v \in V} d_v = 0$.

A multi-terminal network for a single commodity flow can be reduced to a standard two-terminal network by introducing one virtual source node and one virtual sink node. Virtual arcs connect the new source to true sources and true sinks to the new sink. The transit times of these virtual arcs are zero. The capacities of arcs connecting to the virtual source with all other sources are bounded by the capacities of these sources. The capacities of arcs connecting to virtual sink from true sinks are bounded by the capacities of these sinks.

2.2.2 Flow

Suppose $f : A \times \mathcal{T} \to \mathbb{R}_{\geq 0}$ be the dynamic flow function defined on arcs. The value $f_e(\tau)$ it enters to the arc $e = (v, w)$ at time τ arrives at node w at $\tau + \theta_e$. The flow excesses at the node induced by a flow on arcs are denoted by $h : V \times \mathcal{T} \to \mathbb{R}_{\geq 0}$. As a directed static network $N = (V, A, u_e)$ can be obtained by omitting the time components of $N = (V, A, u, \theta, T)$. So, the same function f with time component represents the dynamic flow. The flow excesses $h : V \times \mathcal{T} \to \mathbb{R}_{\geq 0}$ at vertices induced by a flow on arcs is denoted and defined by:

$$
\begin{aligned}
h_v : \quad &= \sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e, \\
h_v(\tau) \quad &:= \sum_{e \in A_v^+,\, \tau \geq 0} f_e(\tau) - \sum_{e \in A_v^-,\, \tau - \theta_e \geq 0} f_e(\tau - \theta_e), \quad \tau \in \mathcal{T}.
\end{aligned}
$$

This function over continuous time is defined by,

$$
h_v(\tau) \quad := \sum_{e \in A_v^+} \int_o^\tau f_e(\mu) d\mu - \sum_{e \in A_v^-} \int_0^{\tau - \theta_e} f_e(\mu) d\mu, \quad \tau \in [0, T),
$$

where the flow function is Lebeague measurable.

Above definition of flow is based on arc flow. In a similar form, a path flow can also be defined. Suppose Γ be the set of all $s-t$-paths where $s \in V^+$, $t \in V^-$, and $\Gamma_e := \{\gamma \in \Gamma \mid e \in \gamma\} \subseteq \Gamma$ denote the subset of paths incident to e. The path flow function $g : \Gamma \longrightarrow \mathbb{R}_{\geq 0}$ satisfies the following capacity constraints

$$u_e \geq \sum_{\gamma \in \Gamma_e} g(\gamma) \geq 0.$$

The path flow value is

$$|g| := \sum_{\gamma \in \Gamma} g(\gamma).$$

The flow conservation is almost obvious in path flow. An arc flow can be computed by adding all the path flows incident to that arc. Construction of a path flow from an arc flow is not that much possible. Though, path flows are dependent on arc flows. The path flow and an arc flow are strongly dependent on each other. Every arc flow can be written in terms of path flows and conversely. Let $\Gamma' \subset \Gamma$ be the $s-t$- path decomposition of an arc flow f together with g_γ for paths $\gamma \in \Gamma'$ where g_γ assumed to be path flow with $g_\gamma = 0$ for all $\gamma \in \Gamma \backslash \Gamma'$. Then, $|f| = |g|$ holds and such induced arc flow on each arc can not exceed the flow value of the original one as the flow can be sent through cycles without changing the value of an edge flow. The cycle flows can be omitted as they are found during the decomposition. For any arc flow f, path decomposition can be completed in polynomial-time, [11]. The path flow satisfies the flow conservation property so that flow travels through the path without any breaks and enters an arc $e = (v, w)$ at time $\theta(\gamma_{[s,v]})$, *i. e.*, after traveling along all its preceding arcs. The transit time of a path is

$$\theta_\gamma := \sum_{e \in \gamma} \theta_e.$$

The collection of $s-t$- paths incident to an arc e at time τ is denoted and defined by:

$$\Gamma_e^\tau := \{\gamma \in \Gamma \mid \theta(\gamma_{_{[s,v]}}) \leq \tau \text{ and } \theta(\gamma_{_{[v,t]}}) < T - \tau\}.$$

Let $\overleftrightarrow{\Gamma}_e^\tau$ and $\overleftrightarrow{\Gamma}_{\overleftarrow{e}}^\tau$ be the set of paths incident to e, and its reverse arc at time τ, respectively, where $\overleftrightarrow{\Gamma}$ denotes the collection of $s-t$-paths in the residual network. Suppose $g : \Gamma \times \mathcal{T} \longrightarrow \mathbb{R}_{\geq 0}$ and $g^c : \Gamma \times \mathbf{T}_c \longrightarrow \mathbb{R}_{\geq 0}$ be the path flow over time with the discrete and continuous time setting, respectively. In both sort of the time settings, flow should satisfy the capacity constraint in their respective domains,

$$\sum_{\gamma \in \overleftrightarrow{\Gamma}_e^{\tau}} g(\gamma, \tau - \theta(\gamma_{[s,v]})) \leq u_e,$$

where $e = (v, w)$. The value of flow over time with discrete and continuous time setting are defined by:

$$|g| := \sum_{\gamma \in \Gamma} \sum_{\tau=0}^{T} g(\gamma, \tau) \text{ and } |g^c| := \sum_{\gamma \in \Gamma} \int_0^T g(\gamma, \tau) d\tau.$$

Note that the flow over continuous time can be obtained by replacing the summation of time component by integration. As in the static, $|f| = |g|$ also holds for flow over time.

2.3 Network Flow Models

The modern network flow problem is motivated by the US military force-driven question, how the Russian railway system could be disconnected completely between the eastern and western parts in the cheapest way, posed by Harris and Ross [53]. The computational complexity of such a disconnection (cut) has been reduced by Ford and Fulkerson [54] by establishing the well known max-flow-min-cut theorem. The theorem states that the value of maximum flow is equal to a minimum cut capacity. The same result is independently established in [55] too. But, a constructive solution procedure was not provided in the first one. Details on the models are presented in following.

2.3.1 Node-arc

Static flow model. The node-arc network flow model presented in [53, 54] satisfies Constraints (2.1-2.3). This is presented by assuming the intermediate conservation.

$$\sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e = \begin{cases} h_v, & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \\ -h_v, & \text{if } v \in V^-, \end{cases} \tag{2.1}$$

$$h_v \geq 0, \; \forall \, v \in V, \tag{2.2}$$

$$u_e \geq f_e \geq 0, \; \forall \, e \in A. \tag{2.3}$$

Out going flow from the sources, conservation of flow in intermediate nodes and entering flow to the sinks are respectively shown in Constraint (2.1). Constraint (2.3) represents bounds of flows on arcs. The objective of the maximum flow problem (MFP) is to maximize

$$\sum_{v \in V^+} h_v. \tag{2.4}$$

If the flow is maximized under the given priority of terminals in a multi-terminal network then it is said to be lexicographically maximum flow (cf. Section 2.4.1). The minimum cost flow problem minimizes the total cost to satisfy the given supplies and demands. The minimum cost flow formulated as in [11] is

$$\min \sum_{e \in A} c_e f_e, \tag{2.5}$$

$$\sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e = d_v, \; \forall \, v \in V, \tag{2.6}$$

$$u_e \geq f_e \geq 0 \; \forall \, e \in A. \tag{2.7}$$

The conservation and capacity constraints are given by (2.6) and (2.7), respectively. Equation (2.8) represents the feasibility of supplies and demands. This constraint implies that the total supply is equal to the total demand,

$$\sum_{v \in V} d_v = 0. \tag{2.8}$$

Dynamic flow model. Discrete dynamic network flow model introduced by Ford and Fulkerson [1] satisfies Constraints (2.9-2.12). Constraint (2.12) ensures that the flow does not enter arc e at time τ if it will have to leave the arc after given time horizon. The maximum dynamic flow that can enter the arc e within each integral time step τ is bounded by the time varying capacity $u_e(\tau)$, this is ensured by Constraint (2.11). Flow conservation conditions are ensured in Constraint (2.9).

$$\sum_{e \in A_v^+} f_e(\tau) - \sum_{\substack{e \in A_v^- \\ \tau - \theta_e \geq 0}} f_e(\tau - \theta_e) = \begin{cases} h_v(\tau), & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \ \forall\, \tau \in \mathcal{T}, \\ -h_v(\tau), & \text{if } v \in V^-, \end{cases} \tag{2.9}$$

$$h_v(\tau) \geq 0, \ \forall\, v \in V, \ \forall\, \tau \in \mathcal{T}, \tag{2.10}$$

$$u_e(\tau) \geq f_e(\tau) \geq 0, \ \forall\, e \in A, \ \forall\, \tau \in \mathcal{T}, \tag{2.11}$$

$$f_e(\tau) = 0, \ \forall\, e \in A, \tau = \overline{T - \theta_e + 1,\ T}. \tag{2.12}$$

Flow value at T is defined in (2.13) and is to be maximized for the maximum dynamic flow problem,

$$\sum_{\tau=0}^{T} \sum_{v \in V^+} h_v(\tau). \tag{2.13}$$

The maximum dynamic flow problem (MDFP) can be transformed into the minimum cost flow problem (MCFP) by assuming the transit time of MDFP as the cost of the MCFP. If the objective of the dynamic network flow problem is to minimize the T to sent the flow from the sources to sinks then it is said to be quickest flow problem. The maximum dynamic flow is said to be the earliest arrival flow if the flow is the maximum for every possible time from the beginning. Moreover, MDFP maximizes the total flow at a given time horizon, but the earliest arrival flow maximizes the flow at each possible time and also minimizes the shipping time. Thus, every earliest arrival flow is the maximum dynamic flow, but converse may not be true. Details of dynamic flow problem (DFP) is given in Section 2.5.

2.3.2 Path-arc

The first network flow model introduced in 1950's considers node-arc flow model. The path-arc network flow model is an extension of the node-arc flow model. The linear programming of path-arc model can be formulated as:

$$\max \sum_{\gamma \in \Gamma} g(\gamma), \tag{2.14}$$

$$\sum_{\gamma \in \Gamma_e} g(\gamma) \leq u_e, \; \forall\, e \in A, \tag{2.15}$$

$$g(\gamma) \geq 0, \; \forall\, \gamma \in \Gamma. \tag{2.16}$$

Due to the exponential number of paths this problem is not polynomially solvable. The problem can be solved polynomially using a seperation oracle. Let x_e be the dual variable corresponding to each arc. Then, the minimum cut problem as the dual of the Problem 2.14-2.16 can be formulated as:

$$\min \sum_{e \in A} u_e x_e, \tag{2.17}$$

$$\sum_{e \in \gamma} x_e \geq 1, \; \forall\, \gamma \in \Gamma, \tag{2.18}$$

$$x_e \geq 0, \; \forall\, e \in A. \tag{2.19}$$

The maximum flow and the minimum cut problems are totally dual integral, [56]. Hence, any solution to one problem will correspond to a solution for others. Here, primal has one variable g_γ corresponding to each path and one constraint for all $e \in A$ as a result dual has only one variable x_e corresponding primal constraint and one constraint corresponding to the primal variable. Thus, the dual problem equals the definition of cut. For this, suppose C be an $s-t$-cut. Since an $s-t$-cut blocks all the $s-t$-paths. The solution of the minimum cut can be obtained by assuming $x_e = 1$ for all $e \in C$. There may be exponentially many paths which lead to the exponentially many variables in primal, and exponentially many constraints in the dual. By decomposition theorem, the optimal solution of the primal only uses a polynomial number of $s-t$-paths with the non-zero flow. The column generation method finds the optimal solution of the primal in polynomial-time. The path- arc flow and path-arc flow over time with a more general path setting is presented in Chapter 3.

2.3.3 Contraflow

Contraflow approach is another emerging and widely accepted model for evacuation planning that increases the outbound road capacities by reversing the direction of roads towards the safe destinations. In contraflow approach, the auxiliary network of given network will be constructed by adding the capacities of two way arcs or paths and allowing the directions in both ways with symmetric capacities and transit times. Construction of auxiliary network with arc and path reversals is as follows.

Arc reversals. The auxiliary network $\tilde{N} = (V, \tilde{A}, \tilde{u}, \theta, T)$ is constructed from given evacuation network N as $\overrightarrow{e} = (v, w) \in \tilde{A}$, if $\overrightarrow{e} \in A$ or $\overleftarrow{e} = (w, v) \in A$. The arc capacity function $\tilde{u}$ is given by $\tilde{u}_{\tilde{e}} = u(\overrightarrow{e}) + u(\overleftarrow{e})$ for all arcs $\overrightarrow{e} \in \tilde{A}$. The transit time $\tilde{\theta}$ is defined as follows

$$\tilde{\theta}(\tilde{e}) = \begin{cases} \theta(\overrightarrow{e}) & \text{for } \tilde{e} \in A, \\ \theta(\overleftarrow{e}) & \text{else,} \end{cases} \qquad \forall\, \tilde{e} \in \tilde{A}.$$

Path reversals. Let $\Gamma = \{\overrightarrow{\gamma}, \overleftarrow{\gamma}\}$ be the set of all paths in contraflow network N with capacities $u(\overrightarrow{\gamma}) = \min\{u_e : e \in \overrightarrow{\gamma}\}$ and $u(\overleftarrow{\gamma}) = \min\{u_e : e \in \overleftarrow{\gamma}\}$. We define the undirected auxiliary network $\tilde{N}$ by adding the capacities on the corresponding two-way paths and keeping the transit time (if any) fixed. The set of arcs and paths are denoted by $\tilde{A}$ and $\tilde{\Gamma}$, where $\tilde{e} \in \tilde{A}$ and $\tilde{\gamma} \in \tilde{\Gamma}$. Then the capacity function is defined as $\tilde{u}(\tilde{\gamma}) = \min\{u_{\tilde{e}} : \tilde{e} \in \tilde{\gamma}\}$ while the transit times (if any) on paths remains the same.

2.4 Flow Problems in Static Network

The first network model motivated by the cold war was formulated to extract the minimum cut to disconnect the network completely. Later, it has been shown that it is the dual problem of the linear programming formulation of MFP. The problem has been solved by applying the simplex algorithm, Dantzig and Fulkerson, [57]. Then, the problem has been solved in [58] showing the equality of the maximum flow value and value of the minimum cut. An algorithm was then developed, though the algorithm was pseudo polynomial. The problem aroused in the cold war was solved by applying heuristic, [59]. Different flow models have been developed after this model, some of the variants are presented in following section. For details on the development of network flow models, we refer to [60].

2.4.1 Flow maximization

The objective of the maximum flow problem is to send as much flow as possible from the particular vertices, say sources, to the particular vertices say sinks, satisfying Constraints (2.1-2.3). This algorithm was developed after the establishment of well known Max-Flow=Min-Cut-Theorem, [51, 55]. Basically, two type of algorithms have been developed to solve the problem. One is based on augmenting path and the next is based on pre-flow push algorithms. Both of the approach also shows the duality. The algorithms based on augmenting path satisfies the feasibility at each steps of algorithms where the pre-flow push based algorithm may violate the feasibility in primary stage, but ultimately, it will reduce the infeasibility. Most of the algorithms use residual network during the execution.

Suppose f is the arc flow defined in $N = (V, A, u, s, t)$. The residual network with respect to the flow f is denoted $\overleftrightarrow{N} = (V, \overleftrightarrow{A}, u_f, s, t)$, where $\overleftrightarrow{A} := A \cup \{\overleftarrow{e} \mid e \in A\}$ represents the set of arcs and $\overleftarrow{e} = (w, v)$ is the reverse arc of $\overrightarrow{e}$. The residual capacity with respect to the flow f is defined as:

$$u^f(e) := \begin{cases} u_e - f_e & \text{if } e \in A, \\ f_e & \text{if } e \in \overleftarrow{A}. \end{cases}$$

The residual network allows algorithms for backward flow which has been passed in earlier iterations. The augmenting algorithm terminates with optimal flow if no more $s - t$-path is possible in the residual network. The termination of the algorithm is possible whenever capacities on the arcs are integral. In the case of irrational arc capacities, algorithm may not terminate, [61]. The algorithm heavily depends on the choice of paths. As a consequence, the complexity of the algorithm $\mathcal{O}(Unm)$ is the pseudo polynomial, where U is the largest capacity of arc. The complexity of the algorithm has been reduced independently to $\mathcal{O}(nm^2)$ by selecting the shortest path in [62, 63]. Again, the blocking flows approach has been implemented to improve the complexity of the augmenting algorithm to $\mathcal{O}(n^2m)$ in [63] whereas $\mathcal{O}(n^3)$, [64, 65].

Goldberg and Tarjan [66] have provided the push-relabel algorithm in which the augmenting path has not been used. The new selection rules and sophisticated data structures have been applied to improve this generic algorithm in [67]. The complexity of the improved algorithm is $\mathcal{O}(nm \log_{\frac{m}{n}\log n} n)$, polynomial-time. Again, Goldberg and Rao [68] improve the complexity to $\mathcal{O}(\min\{n^{\frac{2}{3}}m, m^{\frac{1}{2}}\} \log \frac{n^2}{m} \log U)$. The presented algorithm are efficient in terms of complexity but, not in implementation. The fastest algorithm in terms of implementations and based on push-relabel is given in [69]. The algorithm considered to be one of the fastest implementations of the push-relabel, has improved the performance in practice using several heuristics is given in [69]. But, the complexity remains the same. On the same line, Goldberg [70] has investigated an algorithm, better for some instances and comparably good in general. Till now, the algorithm provided in [71] is considered to be the best algorithm in terms of complexity. Complexity of the algorithm is $\mathcal{O}(nm)$. If the number of arcs is in $\mathcal{O}(n)$, the complexity becomes $\mathcal{O}(\frac{n^2}{\log n})$. All of the algorithm presented above uses residual network. The abstract flow algorithm presented in [72] is the exceptional algorithm where linear program has been used rather than the residual network. This algorithm has been applied even in classical network flows, see in [31].

The above discussed algorithms are for the maximum flow problem considered in two-terminal network but are also true for the multi-terminal network. The multi-terminal network can be reduced to the standard two-terminal network by introducing one virtual source vertex and one virtual sink vertex. Virtual arcs connect the new source to true sources and true sinks to the new sink. The capacities of arcs connecting to the virtual source with all other sources are bounded by the capacities of these sources. The capacities of arcs connecting to virtual sink from true sinks are bounded by the capacities of these sinks.

Lexicographically maximum flow. A prioritized network is a multi-terminal network which consists prioritized terminals. Under the given priority of terminals, two flows can be compared according to departure/arrival flows form/in the sinks or sources. A flow value is said to be lexicographic if it is compared according to the rank of the terminals. Let $N = (V, A, u)$ be a prioritized network with priority $t_1, t_2, \ldots, t_n;\ t_i \in V^+ \cup V^-$. Let

$$|f|_t := \begin{cases} \sum_{e \in A_t^+} f_e, & t \in V^+ \\ \sum_{e \in A_t^-} f_e, & t \in V^- \end{cases}$$

be the out/in flow value from/in the source and sink, respectively. Suppose f^1 and f^2 be the terminal respecting flows, f^1 is said to be lexicographically bigger than f^2 and written as $f^1 \geq_L f^2$ if $\exists\ l \in \{0, 1, \ldots, k-1\} : \forall\ i \in \{1, 2, \ldots, l\} : |f^1|_{t_i} = |f^2|_{t_i}$ and $|f^1|_{t_{l+1}} > |f^2|_{t_{l+1}}$ or $\forall i \in \{1, 2, \ldots, k\} : |f^1|_{t_i} = |f^2|_{t_i}$. The maximum flow respecting the rank of the terminals is said to be lexicographically maximum flow.

The lexicographically maximum flow is firstly introduced in [73]. The arrival and departure patterns in sources and sinks, respectively, have been defined using maximum static flow algorithm. It has been shown that the maximum flow does not depend on the arrival and departure pattern. For any two flows having their arrival and departure patterns, the third flow with the same value can be constructed having the same value with arrival pattern of the one and departure pattern of the other. The lexicographically maximum flow problem having integral arc capacities has been solved in [73].

Author in [74] solves the problem in the single source and multi-sink network having the highest rank to the source and without given ranks to the sinks. A more general parameterized algorithm for the lexicographically maximum flow problem which is based on the maximum flow algorithm, [75]. The complexity of the algorithm is the same as in the chosen maximum flow algorithm which requires the time taken by one maximum flow algorithm. This model is useful in those evacuation scenarios if flow is required to leave some nodes earlier. Thus, the lexicographically maximum flows can be used to compute an earliest arrival flow and a quickest transshipment. Authors in [76] applied this model to compute a quickest transshipment, [77]. The lexicographically minimum cost model has been applied to design evacuation planning problem in [10].

Multi-commodity flow. So far the above network flow models consider the same type of flows called the single-commodity flow that are interchangeable. The multi-commodity network flow considers the different commodities as flows and are not interchangeable to each other. Suppose $f_1, f_2, \ldots, f_k$ be k commodities with $K = \{1, 2, \ldots, k\}$ in multi-commodity network flow model. As flows share the capacity of arc, the capacity constraint,

$$\sum_{i \in K} f_i(e) \leq u_e, \text{ for all } e \in A. \tag{2.20}$$

To send different commodities to the different sinks from the different sources, multi-commodity maximum flow problems must be defined in the multi-terminal network. The objective of the maximum multi-commodity flow problem is to find k network flows f_i for every commodities $i \in K$ satisfying the capacity Constraint 2.20 and the total flow, $\sum_{i \in K} |f_i|$ is maximum. A linear programming formulation can be given to the maximum multi-commodity flow problem, but an efficient algorithm is still lacking due to the size of linear program. An FPTAS has been given in [78]. The approximation is based on the path formulation of fractional network flows. The numerical stability and the huge number of iterations can be problematic in practice though it is reasonably simple to be implemented. It is possible to implement the technique with some modifications for huge instances, [79].

2.4.2 Cost minimization

The feasibility of minimum cost flow problem can be determined by solving a maximum flow problem, [11, 80]. For this, one introduces a super source node s^*, a super sink node t^*, source arcs (s^*, v) with capacities d_v for $v \in V^+$ and sink arcs (v, t^*) with capacities $-d_v$ for $v \in V^-$. Recall that $\sum_{v \in V^+} d_v = \sum_{v \in V^-} d_v$ holds. If the maximum flow saturates all the source arcs, the minimum cost flow problem is feasible; otherwise, it is infeasible.

Let $\zeta : V \longrightarrow N$ be the balances with respect to the flow f. It is defined by

$$\zeta_f(v) := \sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e, \;\; \forall\, v \in V. \tag{2.21}$$

A flow is said to be circulation if $\zeta_f(v) = 0, \text{ for all } v \in V$. The minimum cost circulation can be applied to solve minimum cost flow problem. A flow will be sent through the negative-length cycle whenever there exists negative-length cycles otherwise it will create unnecessary cost. Both of the problems, minimum cost circulation and maximum flow should obey the capacity constraint, in this sense they are similar. In order to satisfy all the demands, flow may also be sent on expensive paths. The minimum cost flow problem is said to be a transshipment if it satisfies demands. The first algorithm for the problem is given in [11], solves the problem in pseudo polynomial-time. Many algorithms are developed after this algorithm, but most of them are pseudo polynomial, [81]. The successive shortest path algorithm is one of them. The complexity of the successive shortest path algorithm is $\mathcal{O}(nm + \mathbb{B}(m + nlogm))$, where $\mathbb{B} = \frac{1}{2}\sum_{v \in V} |\zeta_v|$ [62, 82]. This algorithm is heavily applied in flow over time problem. The first polynomial-time algorithm is developed in [83]. The complexity of the algorithm is $\mathcal{O}(m \log m(m + nlogn))$. The network simplex algorithm has been applied in [84, 85] to solve the problem. A scaling algorithms presented in [86] are considered to be the fast algorithms in practice.

2.5 Flow over Time Problems

As mention earlier, the network flow problem is motivated from real world problem. The problem posed in the cold war has been solved in static network. The static network flow can be applied even in image processing, [87]. But, most of the real life problems consists time which can not be deal with the static network. The dynamic network flow problem considers time as a factor has been firstly introduced and solved in [1, 11]. The term "dynamic" has also been used if the input of the network changes with respect to time, [88]. The difference between the dynamic networks from static networks is that the dynamic networks have transit time τ_e for each arc e. Thus, the network flow problems with time factor are known as dynamic network flow problems. Basically, there are two type of time setting, namely discrete and continuous. As in static network, flow over time problems can be categorized according to their objectives. Here, the objectives are taken as: the flow maximization, time minimization, and the both, that is, flow maximization and time minimization in different network topologies. The flow maximizing problem maximizes the flow value at given time horizon whereas, time minimizing problem minimizes the transshipment time of flow to satisfy the demands and supplies. Moreover, one can add cost component, constraint relating cost and seeks to minimize the overall costs as well. Using time-expanded network, the flow over time problem can be transformed into the static network flow problem.

The first model considers the discrete time setting, [11], where a unit of flow entering an arc at τ arrives at the head by traveling the whole arc at $\tau+\theta_e$. Later, the continuous time model has been developed as a generalization of discrete model, [89]. Consideration of time varying capacity and transit time in the dynamic network makes the problem dynamic. If the properties of arcs change according to the flow units it is said to be load-dependent and it becomes the commodity-dependent if the changes is due to the transship commodity. For time-dependent arc capacities model, time window should be fixed. Let $l > 0$ be the time window, then the capacity constraint can be defined by

$$\int_{\theta}^{\theta+l} f(e,\mu)d\mu \leq U_e,$$

where $U_e \in R_{>0}$ is the upper bound of aggregated arc capacity. This can be applied while using the bridge in the network model. The network flows with such capacities is said to be bridge flows. Author in [90], firstly investigated the concept of aggregated arc capacity and based on this approach a FPTAS has been developed in [91]. Flows over time problems have the extensive application on real-world. Its application in traffic flow modeling are considered in [92, 93, 94]. It has been used in for supply location selection and routing in large-scale emergency material delivery in [95, 96] and the model with logistics is presented in [97, 98]. Similarly, [99] uses it for scheduling; [100] uses for cloud data storing; and [101] uses for network structuring. For more details, see in [102, 103].

Discrete vs continuous time. For constant transit time, Fleischer and Tardos [12] provided a Transformation (2.22), also known as the natural transformation, which transforms the continuous time dynamic flow model into the discrete time setting. The basic difference in these two time settings is that the flow which is entering e at $\tau - \theta_e$ arrives at the head of e at time τ for discrete but at time $[\tau + 1)$ in the continuous time setting. The transformation can be represented as

$$f(e, \tau) = \int_0^{\theta_e} f(\tau - \mu) d\mu = \sum_{\mu=0}^{\theta_e - 1} f(\tau - \mu). \tag{2.22}$$

Using flow decomposition, they have shown that the optimality remain the same in both of the time settings. The network flow model has developed in different dimension by defining the Borel flows, which is relying on the Borel measure, [104].

Time-expanded model. The time-expanded network is the static network constructed from the dynamic network. It is constructed by introducing copies of the network topology to certain point of time. Let $N = (V, A, u(\tau), \theta, T)$ be a dynamic network, the corresponding time-expanded network is denoted by $N_T = (V_T, A_T, u_T)$, which is a static representation of the dynamic network. Construction of time-expanded network of dynamic network for the discrete time setting is as follows:

$$V_T := \{v(\tau) : v \in V, \tau \in \mathcal{T}\}, \tag{2.23}$$

$$A_T := \{(v(\tau), w(\tau + \theta_e)) : e = (v, w) \in A, 0 \leq \tau \leq T - \theta_e\}, \tag{2.24}$$

$$A_T^H := \{(v(\tau), v(\tau + 1)) : v \in V, \;\; 0 \leq \tau \leq T - 1\}, \tag{2.25}$$

$$u_T^{e(\tau)} := u_e(\tau), \text{ for } e(\tau) \in A_T, \tag{2.26}$$

$$u_T^{e(\tau)} := \infty, \text{ for } e(\tau) \in A_T^H. \tag{2.27}$$

Equations (2.23), 2.24, and 2.26 represent set of vertices, set of movable arcs, and capacity of movable arcs in the time-expanded network, respectively. Similarly, Equations 2.25 and 2.27 represent set of the holdover arcs and its capacity, respectively. Similarly, the time-expanded network can be created for the dynamic network with the continuous time setting. An example of time-expanded network corresponding to Figure 1 for the continuous time setting with $T = 4$ is shown in Figure 2. The time-expanded network with discrete time setting can be obtained by taking only the lower limits of the intervals as the time point. Let $(v(t), w(t + \theta_e)) \in A^T$ and let $f_e(t)$ be a flow in the dynamic network $N = (V, A, u(t), \theta, T)$. The corresponding flow function

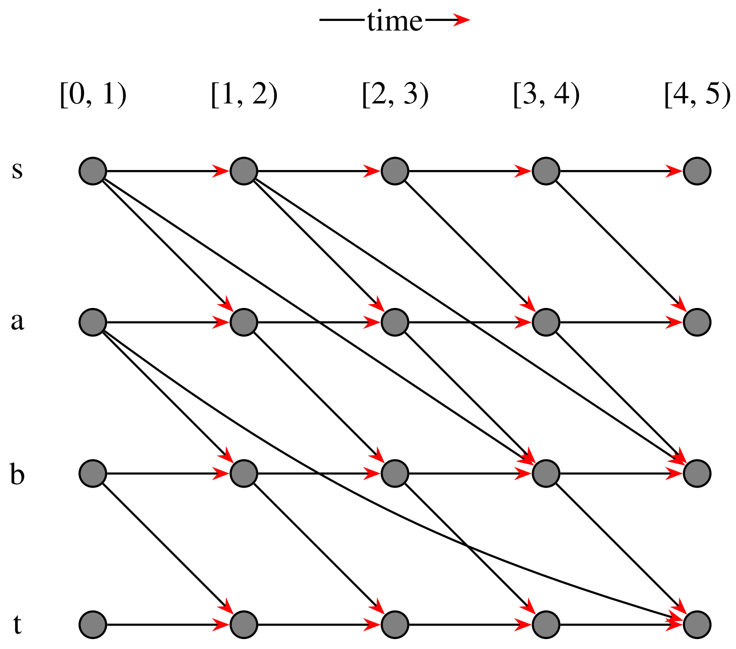


Figure 2: Time expansion of Figure 1.

in the time-expanded network $N_T = (V_T, A_T, u_T)$ is defined by

$$f_T^{e(\tau)} := f_{(v(t),w(t+\theta_e))} = f_e(t) \ \ \forall \, e(t) \in A_T. \tag{2.28}$$

Relation (2.28) is a bijection from the set of feasible flows in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$ onto the set of feasible flows in the time-expanded network $N_T = (V_T, A_T, u_T)$ so that the dynamic flows $f_e(t)$ with time horizon T are equivalent to the static flows $f_T^{e(t)}$ in the time-expanded network, [105]. From this construction, it can be concluded that the maximum dynamic flow can be computed in the time-expanded network and the optimality of the flow remains the same in both network. The size of the time-expanded network depends on T. The size of the time-expanded network is not necessarily polynomial in the given input size problem, as T is not bounded. Hence, any algorithms based on the time-expanded network are pseudo-polynomial.

Condensed network. A condensed time-expanded network N_T^{Δ} is constructed by rescaling the time by the factor $\Delta > 0$. The time horizon is scaled by $\lceil \frac{T}{\Delta} \rceil$, where $\lceil . \rceil$ is the ceiling function. This implies that, the transit times are multiple of Δ. The set of vertices in the condensed time-expanded network is denoted and defined by $V_{\rho\Delta} := \{v_\rho \mid v \in V \text{ and } \rho = 0, \dots, \lceil \frac{T}{\Delta} \rceil\}$, relates to flow though V in $[\rho\Delta, (\rho+1)\Delta)$. Any dynamic flow problem with time horizon is equivalent to the static flow of equal cost in N_T^Δ whenever $\lceil \frac{T}{\Delta} \rceil$ is an integer. Otherwise, the equal value will be possible before $T + \Delta$. The approximate solution to the given network can be obtained by rounding up the transit times in N_T^Δ. As a result, optimality gap will be minimum, [106].

Fan and bow networks. Suppose θ^s be the integer valued non-decreasing left-continuous inflow-dependent transit times step function. Then the fan network introduce in [93] is denoted and defined by $N_T^F = (V_T^F, A_M^F \cup A_H^F)$ with

$$V_T^F \;:=\; \{v_\tau \mid v \in V,\ \tau = 0, 1, \ldots, T\}, \tag{2.29}$$

$$A_H^F \;:=\; \{(v_{\tau,}, v_{\tau+1}) \mid v \in V,\ \tau = 0, 1, \ldots, T-1\}, \tag{2.30}$$

$$A_M^F \;:=\; \{(v_{\tau,}, w_{\tau+\theta_e^s(\tau)}) \mid e = (v, w) \in A,\ \tau = 0, 1, \ldots, T-1, \}. \tag{2.31}$$

Equation 2.29, 2.30, and 2.31 represent set of vertices, holdover arcs, and flow movable arcs, respectively. The capacity of holdover arcs is infinite and the flow value will be controlled by the capacity of flow movable arc. The fan network has been applied to deal with the inflow-dependent transit time problem.

Due to the exponential size of time-expanded fan network, it is difficult even to approximate the solution for time varying attributes in fan network. Again, author in [93], introduce bow network to deal with certain inflow-dependent transit time problems. For this, define breakpoints of flow rate by $0 < z^1 < z^2 < \cdots < z^k = u_e$ with the corresponding transit times $\theta^1 < \theta^2 < \ldots \theta^k$, respectively. Then the bow network is denoted and defined as $N^B = (V^B, A^B)$ with $V^B = V$. The set of arcs in bow network consists two types of arcs, namely, bow and regulating arcs. They are denoted by u^i and $r^i,\ i = 1, 2, \ldots, k$, respectively. The bow arcs have infinite capacities with transit time θ^i and the regulating arcs have capacity z^i and zero transit times. Indeed, a time-expansion of bow network is the fan network. The dynamic flow problem with inflow-dependent transit time can relate with the dynamic flow in bow graph by

$$f_a^B(\tau) := \begin{cases} f_e(\tau), & \text{for} a = b^i \text{ or } a = r^j\ j \geq i, \\ 0 & \text{otherwise.} \end{cases}$$

Based on above transformations, it has been shown that, every dynamic flow with inflow-dependent transit time in N can be considered as a dynamic flow with constant transit times in bow network, but converse may not be true.

Standard chain decomposition. The temporally repeated solution approach of [11] has been applied in most of the polynomial-time dynamic flow algorithms. The maximum flow minimum cost flow has been decomposed into chain flows by assuming the transit time as costs. Let $\Gamma := \{\gamma_1, \gamma_2, \ldots, \gamma_p\}$ be the set of such chains having the value $val(\gamma_k)$. Then the static solution can be expressed as $f = \sum_{k=1}^{p} \gamma_k$. The standard chain decomposition repeats the chain flows in the same direction as that of static flow. This decomposition can be used to define the dynamic flow by repeating the flow at each time from time zero to $T - \theta_\gamma$. The obtained dynamic flow from the chain decomposition Γ_T has the value

$$val(f, T) = \sum_{\gamma \in \Gamma} (T - \theta(\gamma) + 1) val(\gamma) = (T + 1) val(f) - \sum_{e \in A} f_e \theta_e. \tag{2.32}$$

Note that the chain decomposition of the flow is not unique and the flow value does not depend on the choice of chain. The solution of a two-terminal dynamic network can be obtained in polynomial-time by solving the minimum cost circulation problem. But, this approach can not be applied to solve the earliest arrival flow and dynamic flow in multi-terminal network. The standard chain decomposition is useful for approximation.

Non-standard chain decomposition. A non-standard decomposition introduced in [77, 107] may not repeat the flow but may cancel the flow value, if possible. To cancel the flow, feasibility of the cancellation should be checked. Suppose $e = (v, w)$ lies in the chains $\gamma, \gamma' \in \Gamma$ as forward and backward arcs. To cancel the flow, the amount of flow on γ from source to v that need to be canceled on e have to reach v before γ' reaches there. Moreover, the flow from w to the sink through γ that has to be canceled by γ' should not already left w. The optimality of dynamic flow will not be affected. Any flow obtained from the non-standard chain decomposition is called generalized temporally repeated flows.

2.5.1 Flow maximization

The maximum dynamic flow problem was introduced by Ford and Fulkerson [1, 11]. The objective of the problem is to send as much flow as possible within a given time horizon. An extended dynamic network can be defined by adding virtual vertices and arcs as in static network with zero transit times for virtual arcs. Thus, the maximum dynamic flow problem can be reduced to the two-terminal network maximum dynamic flow problem. Basically, there are two solution approaches for the problem. One is to transform the dynamic network into time-expanded and solve the problem using any available algorithms for static network. The other approach reduces the maximum dynamic flow problem into the minimum cost flow problem in an extended network by considering the transit time as the cost. Obviously, the first one takes pseudo polynomial-time, whereas the next solves the problem in polynomial-time.

The maximum dynamic flow with the discrete time setting can be expressed as a linear programming. The flow conservation for each T makes the size exponential. As in path-arc, an equivalent linear programming formulation for the maximum dynamic flow problem can be formulated. Due to the number of paths it will not be smaller and $\mathcal{NP}$-hard to solve. In both of the formulations, flow value should be specified on each arcs and at each point of time. The database of the problem depends on time horizon, increases exponentially. But, this can be done in polynomial size by enforcing the constraint that, flow along the paths should be constant for most of the time point and can switch the flow value of paths polynomially. Such flow is said to be temporally repeated flow, it takes the polynomial steps.

Suppose $N = (V, A, u, \theta, T)$ be a dynamic network and let g be a path flow in the corresponding static network. The temporally repeated flow corresponding to the dynamic network can be defined by

$$f(e, \tau) := \sum_{\gamma \in \Gamma_e^\tau} g(\gamma)$$

for each arc $e = (v, w)$ and $\tau \in \mathcal{T}$ or $\mathbf{T}_c$. The flow in the residual network can be decomposed into path flows using paths $\gamma \in \overleftrightarrow{\Gamma}$. This decomposition is said to be the generalized path decomposition. For a static path flow g be the general path decomposition with $\gamma \in \overleftrightarrow{\Gamma}$, the generalized temporally repeated flow with respect to g can be written as

$$f(e,\tau) := \sum_{\gamma \in \overleftrightarrow{\Gamma}^{\tau}_{e}} g(\gamma) - \sum_{\gamma \in \overleftarrow{\Gamma}^{\tau}_{\overleftarrow{e}}} g(\gamma).$$

The temporally repeated flow is constructed from the feasible static flow, is feasible. In the same line, a dynamic path flow over continuous time can be defined by $g'(\gamma,\tau) = g'(\gamma)$, $\tau \in [0, T - \tau_\gamma)$. But, it may not be valid for generalized temporally repeated flow. The flow value may be negative if flow is reduced by flow value on the backward arc before having any flow on its forward arc. Thus, feasibility should be checked before creating generalized temporally repeated flows.

A temporally repeated flow having the maximum value for the maximum dynamic flow problem with non-negative transit times can be obtained efficiently, [1]. For this, a path γ having contribution $\tau_\gamma - T = -(T - \tau_\gamma)$ for the minimum cost circulation will be decomposed. As a result, the path having smaller transit time may contribute more. The holdover of the flow at intermediate vertices is not required for the optimal solution of the maximum dynamic flow problem. The minimum cost commutation minimizes the value. Thus, the time complexity of minimum cost flow algorithm dominates the complexity of the temporally repeated flow algorithm, and is strongly-polynomial. This algorithm works on extended network. It consists only few virtual arcs so that size of the network may not change significantly. The complexity is still the best. In this regard, it is an ideal algorithm for the maximum dynamic flow problem.

Lexicographically maximum flow. As mention in preceding section, the lexicographically maximum flow problem has been defined and solved in [73]. Its dynamic version consists transit time and fixed time horizon. One of the solution procedure for the lexicographically maximum dynamic flow problem is to convert the dynamic network into the time-expanded network and solve the problem. But, this approach depends on the time horizon, so it is pseudo polynomial. The first polynomial-time algorithm is due to [76, 107]. Their algorithm successively works on residual network to find minimum cost flow in each iteration. Then, decompose such minimum cost flow into chain flow. As the minimum cost flow should be calculated in each iteration, the complexity of minimum cost flow dominates the complexity of this algorithm. Hence, the complexity of the algorithm is $\mathcal{O}(kg(nm))$, where $g(nm)$ represent the complexity of the minimum cost flow. The first evacuation planning problem for building evacuation presented in [10] has been based on lexicographic approach. They have used lexicographically shortest augmenting path in the algorithm. One of the objective of their modeling is to control the unnecessary movement during the evacuation within the building.

Multi-commodity flow. The multi-commodity flow over time is an extension of static multi-commodity flow. This model is more realistic for modeling an evacuation problem having different commodities though the problem is weakly $\mathcal{NP}$-hard due to multi-commodity, [108]. The solution of the problem can be categorized into two approaches, having intermediate storage capability and not. For the case of the single-commodities, solution remain same in the both, but the maximum flow value may be different in multi-commodity. Author in [31] computes a maximum dynamic multi-commodity flow without considering storage. Authors in [109] present a FPTAS to approximate the optimal solution for the maximum dynamic multi-commodity flow without assuming the storage. The problems with possible negative transit time in the single-commodity flow have the same issues of storage in the optimality, [31]. Under certain conditions, the existence of earliest arrival multi-commodity flow is shown in [31]. The non-existence of the earliest arrival multi-commodity flow has been given in [110] . Thus, only the approximation is possible for the problem. The dynamic multi-commodity problem with fractional multi-commodity has been studied in [108].

2.5.2 Cost minimization

The dynamic minimum cost flow problem with cost minimization objective for provided time horizon and the quickest flow problem under given total cost have been considered in [111, 112]. The dynamic minimum cost flow problem with infinite time horizon maximizes the throughput, [113]. Authors in [114] have developed dynamic minimum cost flow problem with time varying attributes. Non of these problems are polynomially solvable, are $\mathcal{NP}-$hard. Thus, only the approximation is possible unless $\mathcal{P} = \mathcal{NP}$. The network flow problem with time minimization objective is known as the quickest flow problem. Here, the transit time of flow on arcs are considered as a cost. For the given amount of flow, this problem finds the minimum clearance time to send the flow from the source to the sink. Thus, the maximum dynamic flow problem can be assumed as its inverse. The quickest flow approach has been applied to find the evacuation time, [115].

Based on the linear fractional programming problem [48], authors in [116] solve the problem in the two-terminal network with complexity $\mathcal{O}(m^2 \log^3 n(m+n \log n))$. Their algorithm is also based on the maximum dynamic flow, solves the maximum dynamic flow problem in each iteration. An improved algorithm for the problem has been given in [117], which is also based on the linear fractional programming problem. The algorithm solves the problem in $\mathcal{O}(n^3 \log(nC))$, C is the largest transit time on arcs. A cancel-and-tighten algorithm has been applied to solve the quickest flow problem in strongly polynomial-time in [118]. This algorithm has the complexity $\mathcal{O}(nm^2(\log n)^2)$ which is the better comparing to the algorithms presented in [116] and [117].

The general quickest flow problem has been solved in [76, 77]. Their algorithm works in time-expanded network and uses non-standard chain decomposition. As a result, the complexity of the algorithm depends on $\log T$, a pseudo-polynomial complexity. The quickest flow problem has also been studied considering the given node-arc capacities and transit time in multi-terminal network. The quickest transshipment has been solved by considering submodular function minimization oracle in [76]. The presented algorithm solves the problem in $\mathcal{O}(k^2\alpha(mn)\log(nTU))$, where $\mathcal{O}(\alpha(mn))$, k, and U denote the the complexity of minimum cost circulation, number of terminals and the largest arc capacity, respectively.

As discussed earlier, the dynamic multi-commodity problem is $\mathcal{NP}-$hard. Only the approximations have been developed for the quickest multi-commodity flow problem with different considerations. The quickest minimum cost multi-commodity flow has been presented in [119]. An FPTAS for the multi-commodity quickest flow problem flow has been presented in [106] by considering storage. They have also provided a

$(2+\epsilon)-$ approximation algorithm for the problem without considering the storage. An FPTAS for the quickest multi-commodity flow with inflow-dependent transit time has been provided in [108].

2.5.3 Flow maximization and time minimization

Shortly after the introduction of the dynamic flow by [1], the existence of the earliest arrival flow in two-terminal network is shown in [120]. The objective of the earliest arrival flow problem is to maximize flow at every possible time point from the beginning. The earliest arrival flow not only maximizes the flow, it also minimizes the evacuation time. Thus, earliest arrival flow implies the quickest flow but converse may not be true and the same for for maximum dynamic flow. If the flow satisfies earliest arrival as well as the latest departure properties such flow is said to be universally maximum flow. If there exists earliest arrival flow in any network then the latest departure flow can be obtained in its reverse network. A reversed network can be obtained by reversing the orientation of arcs of the network. In this sense, the earliest arrival flow can also be considered as the universally maximum flow. This model can be applied to design evacuation planning problem as the most important task is to shift evacuees out of an endangered area as quickly as possible.

The earliest arrival flow algorithms presented in [73] and [121] makes the use of successive shortest path algorithm of [11]. They have used flow cancellation property in generalized path rather than the non-standard chain decomposition. Author in [73], uses the maximum flow algorithm in each iterations. So that, T-times the maximum flow algorithm should be applied. As a consequence, the algorithm becomes pseudo polynomial. A minimum cut has been constructed in time-expanded network to extract the solution in [121]. It has been also recommended to make the use of secondary storage in the paper. Author in [107] makes the use of chain decomposition to solve the problem but the complexity is not better than the algorithms given in [73] and [121]. The complexity of mentioned algorithm is pseudo polynomial-time. The capacity scaling approach has been applies to approximate the solution in fully polynomial-time, [77, 107]. The complexity of the $(1+\epsilon)-$approximate algorithm is $\mathcal{O}(\frac{m}{\epsilon}(m+n\log n)\log U)$, there U is the largest capacity of arcs and ϵ is the fixed error factor.

The earliest arrival flow with time varying capacities and transit times has been studied in [122]. The earliest arrival for multi-terminal does not exist, [123, 124]. It does not exists even in the network having two sinks and one source but can be defined in the network with depth 2, one source, and multi-sink. Based on the lexicographically maximum flow in time-expanded, the existence has been shown in [106]. And the approximate solution under arbitrary transit time is provided in [119]. Similarly, authors in [125] and [126] studied the problem with different considerations. The transshipment problem with earliest arrival property does exist in the network having single sink with fixed supplies on the sources, [123]. Authors in [124, 127] solves the earliest arrival transshipment problem in polynomial-time taking zero transit times on arcs. Using the condensed time-expanded networks, an approximation for the earliest arrival transshipment with arbitrary transit times has given, [106]. They have also presented a pseudo polynomial-time algorithm for the problem with time varying supplies and capacities.

2.6 Contraflow Problems

Increasing number of disasters demand efficient evacuation planning. No movement of the evacuees towards the dangerous places from the safe places because of which the corresponding lanes are empty and its opposite sides are congested. A contraflow is an approach that reserves such unused roads to make traffic smooth. As a result, the flow value will be increased and the egress time can be decreased. The partial contraflow approach reverses only the unused arcs that can be used by the flow. As a consequence, some roads can be saved. Such saved roads can be used for emergencies and logistics.

All-links and fastest-links algorithms developed in [128] are designed to support smart traffic, as the first contraflow algorithms. The all-links algorithm visits all available streets only once starting from the source and minimizes congestion whereas the faster-links algorithm force to use the fastest path from the source. But, they are not assuming fixed capacity and flow so that they can not be applied in the network having fixed attributes and flow. Authors in [129] identified the planning problems in hurricane Katrina and Rita and also criticized the unplanned contraflow ordered and failure to use all the contraflow lanes. The minimum cost flow problem has been solved by applying the flip high flow arc heuristic in a time-expanded network. This heuristic records the flow history and change the orientation of each arc to achieve better solution, [130]. The integer programming formulation has been given with their heuristics and proved the hardness of the contraflow, [13]. Their heuristics improve at least 40% evacuation time by reversing at most 30% of total arcs. The contraflow model and repair of road segments have been solved simultaneously in a multi-model evacuation problem, [131]. The setup time for contraflow operation has been investigated considering lexicographic order in a relaxed contraflow model, [100]. The root choice opportunity is given to evacuees to preform complete contraflow reconfiguration by ignoring background traffic, [132].

The first analytical solution for the contraflow is provided in [15, 16]. They have introduced and solved the maximum contraflow and maximum dynamic contraflow problems by reversing the arcs at time zero (remain reversed for each time). The reversal cost has not been considered in their problems. Their maximum dynamic contraflow problem (MDCFP) considers the discrete-time setting. The auxiliary network has been used to get the solution and decide the reversals in their algorithms. The hardness of the general contraflow problem has been shown in [13, 16], is $\mathcal{NP}$-hard. The MDCFP in a multi-terminal network is $\mathcal{NP}$-hard . Taking the records of arcs whether they have been reversed or not in each time and the decision for the reversals transform the problem into $\mathcal{NP}$-complete. The hardness can be shown by using PARTITION and 3-SAT.

The algorithm presented for MDCFP uses a temporally repeated flow algorithm to extract the optimal solution and remove the cycle flow after flow decomposition. Based on this flow, the algorithm reverses the necessary arcs. The complexity of the algorithm is $\mathcal{O}(h_2(n;m) + h_3(n;m))$, where $h_2(n;m) = \mathcal{O}(nm)$ and $h_3(n;m) = \mathcal{O}(n^2m^3 \log n)$ are the time required for the flow decomposition and the maximum static flow computation, respectively, is polynomial. The lexicographically maximum dynamic contraflow problem (LMDCFP) is introduced to maximize the contraflow value respecting the given rank of the terminals within the given time horizon. This problem has been solved optimally in polynomial-time, [19]. Their algorithm is based on minimum cost circulation. The complexity of the algorithm is $\mathcal{O}(\delta(m \log n)(m + n \log n)))$, where $\mathcal{O}((m \log n)(m + n \log n)))$ denotes the complexity of the minimum cost flow problem in the residual network and δ is the number of iteration.

The quickest contraflow problem minimizes the evacuation time to shift the given amount of flow from the sources to the sinks, [15, 16]. A polynomial-time has been given to the problem in a two-terminal network. The binary search has been applied in the interval of bound on time and compute the maximum dynamic contraflow until the arrival of the last flow at the sink. The hardness of the problem in a multi-terminal network is also shown, harder than PARTITION and 3-SAT. A flow problem is known to be transshipment if it satisfies all the flow demands in the network. For the given supplies and demands, the quickest transshipment contraflow problem has been introduced and solved polynomially, [22]. The quickest contraflow problems with constant and load-dependent transit times have been introduced and solved, [133], with a computational experiment. The multi-commodity quickest contraflow problem has been introduced and an approximation algorithm for the problem has been given in [134].

The maximum dynamic flow maximizes the flow value at a given time point. If the objective is to maximize the flow value at every possible point of time then the problem becomes the earliest arrival flow. The earliest arrival contraflow problem (EACFP) in two-terminal network has been investigated in [17, 19, 135, 136]. They have shown that the EACFP can be solved in pseudo polynomial-time. The same problem in two-terminal series-parallel network has been solved in [17, 136]. The running time of their algorithm is $\mathcal{O}(nm + m \log m)$, polynomial-time. A polynomial-time approximation solution for the EACFP is due to [18], with the complexity $\mathcal{O}(m\epsilon^{-1}(m+n \log n) \log U)$, where $\epsilon > 0$ and U is the maximum capacity of the network. Authors in [22] have solved the EACFP separately in multiple sources and multiple sink networks and also provided an approximation algorithm for the problem in a multi-terminal network.

A transshipment problem is considered to be the earliest arrival if it satisfies all the demands as early as possible from the beginning. Authors in [22] have introduced and solved the earliest arrival transshipment contraflow problem (EATCFP) in both multi-source or multi-sink networks polynomially. Though, the problem in the multi-sink network considers zero transit time on each arc. Moreover, approximation algorithms with pseudo polynomial and polynomial-time complexity have been given to the multi-terminal network with arbitrary transit times and zero transit time, respectively. Note that every earliest arrival transshipment is quickest but not conversely.

Flow over continuous time problem can be transformed into the discrete-time setting using the natural transformation presented in [12]. Authors in [20, 21, 137], have used the natural transformation to transform their continuous-time contraflow models into discrete and solved their problems using the available algorithms for the corresponding discrete models. As the transformation can be completed in linear-time, the complexities of their algorithms remain the same as in discrete models. Authors in [14] have introduced the path reversal contraflow models in abstract network and this approach has been applied to define and solve abstract contraflow models with different objective and topologies in [26, 28, 30]. Chapter 3 presents details on abstract contraflow. The partial contraflow reverses only the necessary arc capacities. The contraflow based on partial lane reversals has been developed in [25, 28]. They have extended and solved different contraflow problem to partial contraflow. The complexities of their contraflow are the same as in contraflow and without contraflow models. To deal with loss and gain of flow value during the evacuation, the maximum dynamic contraflow and earliest arrival contraflow problem in loss network have been introduced and solved in [23, 24]. The network facility with the contraflow approach is extensively discussed in Chapter 4. Furthermore, contraflow with switching cost is given in Chapter 5. Some of the application based on the contraflow approach is presented in the succeeding section. In recent work, the partial contraflow has been applied in a fuzzy network, [138].

2.7 Application in Evacuation

Research on network design and disaster management plays a vital role to effectively protect the lives of people and their surroundings including their infrastructure. The disaster management, disaster relief operation, and humanitarian logistics are carried out with a rich application of mathematical techniques to address the coordination among humanitarian activities concerning vehicle routing, demand forecasting, or the optimization of the resources at various stages including mitigation, preparedness, response, and recovery. It requests for inter-disciplinary research to a mathematician from various fields including social, managerial, medical, engineering, economic, environmental studies, and the applied sciences for the effectiveness of the pre and post-disaster operations management.

In the evacuation planning problems based on the network flow formulations, the flow movement can be categorized as pedestrians, auto-based, or transit-based. Evacuees in the auto-based and transit-based systems are considered as the high and low-mobility populations, respectively. The auto-based evacuees are supposed to clear the hazardous area by using their own vehicles whereas, in the transit-based system, they are to be sent to the transit hubs for further evacuations. In large cities of the developing countries, the majority of the population does not have their own vehicles for the evacuation or are unable to be evacuated themselves. In some situations, it demands special attention due to their ages, language inefficiencies, different health problems, or other physical disabilities. In most of the disaster situations, the great loss is due to a lack of proper planning of such transit-vehicles for such transit-dependent people rather than the disaster itself. On the other hand, evacuation models can be classified into two broad categories as the microscopic and macroscopic. In microscopic, individual parameters like walking speed, physical ability, and the reaction among each individual during their movements are also considered and are based mainly on the simulation approach. But in macroscopic, only their common characteristics are considered such that the group of evacuees be a unit. These two models can also be combined as a sandwich model such that the output of both remains stable. A large variety of solution strategies were developed to address the different aspects of transit-based evacuation planning problems. For details we refer to [4, 5, 103, 139], and the references therein.

Bus-based evacuation planning (BEP) problems are the fundamentals of the transit-based evacuation. Bish [140] has presented the most famous model for the BEP to minimize the time of evacuation in case of short notice using a given number of homogeneous buses satisfying all evacuee demands. In this case, the number of evacuees at the demand nodes might be greater than the capacity of a bus and it demands the split delivery within the pickup locations. But, in the BEP proposed by the authors in [141], the number of evacuees at every source nodes is taken as the integral multiple of bus-loads which does not request for the split delivery. Such type of the deterministic BEP is extended to the robust formulation by the authors in [142], where the exact number of evacuees is not known in advance, although a set of possible scenarios is provided.

Based on the above-mentioned BEPs, Pyakurel et al. in [143] have presented different findings on the solution of transit-based evacuation planning problems and opened a wide horizon of research towards it. In their approach, evacuees were supposed to be gathered themselves from their residents, and then they are evacuated by the transit-buses having homogeneous capacities. In a recent work by Adhikari et al. as in [144], an integrated solution approach is presented for the time minimization evacuation planning problem. The collection of evacuees is in the primary sub-network and will follow the earliest arrival flow pattern, which will maximize the flow of evacuees at each instance from the beginning. Then, such evacuees collected are assigned to the transit-vehicles in a dominant vehicle assignment approach in the secondary sub-network of the integrated system to have the minimum clearance time. The arrival of evacuees is also incorporated with partial arc reversal capability. The prioritized integrated network for the quickest transshipment of evacuees in the minimum clearance time has been considered in [145].

The application of contraflow is not only limited to evacuation planning but also in traffic planning that reduces congestion and traffic jams during the day-to-day office hours, some accident management cases, or some street exhibitions. Various mathematical models, heuristics, optimization, and simulation techniques taking into account macroscopic and microscopic behavioral characteristics deal with contraflow for this transportation network, however, an acceptable solution is lacking due to very high computational costs.

Based on the logistic aspects of the problem, authors in [146] have presented a systematic survey about the contributions concerning the relief distribution networks in response to disaster management with a compact overview of both the theoretical developments as well as the practical applications. Colson et al. [147] presented a list of applications including congestion management, network design, and the management of hazardous materials during an emergency. A network design, capacity planning, and vehicle routing strategy to minimize the fixed operating cost for location allocations and the transportation cost for the vehicle routing is also investigated, [130].

The facility location-allocation models and their solution strategies have been presented in [41]. It is applicable to locate the appropriate facilities in the evacuation network to achieve an efficient solution for the evacuation planning problem. If some road segments blocked due to disasters or by different incidence during the evacuation then a model with alternative route choice to clear the traffic has been investigated, [148]. They have presented a prepossessing algorithm to update the network concerning the real-time traffic information like incident time, incident end time, the severity of congestion, etc. For such alternative route selections, they have developed a multi-commodity network flow optimization model for alternative paths and the corresponding flows. Such formulation becomes mixed-integer nonlinear programming and has been relaxed to a linear formulation that gives better computational results. For the clearance of the blocked road segments after a disaster and to reconnect the road network to the post-disaster activities authors in [149] have presented a new arc routing problem.

In real practice, the Monticello, Minnesota region was evacuated by using the lane based contraflow and crossing elimination strategies, simultaneously. The experiment was conducted with fix number of terminals and full lane reversal of transportation network,

[150]. In the same region, a bi-level model was used to solve the Monticello nuclear plant evacuation problem with contraflow at road segments and crossing elimination at intersection jointly, [151]. The bi-level includes the lane-based network optimization and simulation models. A case study was done for a super typhoon on an evacuation network using the integrated contraflow approach, [152]. A multi-modal integrated contraflow model is solved for uncertain arrivals of evacuees in the region with low mobility population that has little access to personal vehicles, unable to drive due to age, sickness, or any other reasons. In such strategy, the transit-based evacuees and the auto-based evacuees will be evacuated to different destinations. The transit-based models are initiated with vehicle routing problem whereas the integrated strategy contains non-contraflow to shorten the strategy setup time, full-lane contraflow to minimize the evacuation network capacity, and bus contraflow to realize the transit cycle operation. Similarly, a bi-level model integrating lane based reversal design and routing with crossing conflict elimination at intersection to minimize the total evacuation time is also applied during an evacuation, [37].

CHAPTER 3

CONTRAFLOW IN ABSTRACT NETWORK

The abstract flow, generalization of classical network flow into abstract setting deals with the flow paths (routes) satisfying the switching property. The contraflow approach has been embedded with it to introduce abstract contraflow models. Our results increase the flow values and minimizes the time by reducing crossing conflicts with arc reversals toward the safe destinations in evacuation network. Section 3.2.1 presents the maximum abstract contraflow problem and its solution (cf. Problem 3.1, Algorithm 1, Theorem 3.2, and Corollary 3.1). This model has been extended into abstract contraflow over time and solved efficiently, Section 3.2.2 (cf. Problem 3.2, Algorithm 2, Theorem 3.3, and Corollary 3.2). To maximize the flow at every time point, the earliest arrival abstract contraflow has been solved, Section 3.2.4 (cf. Problem 3.4, Algorithm 4, and Theorem 3.5). It is also extended and solved in multi-terminal network considering supplies and demands in Section 3.2.5 (cf. Problem 3.5, Algorithm 5, and Theorem 3.6). For given priority of terminals, the lexicographically maximum abstract flow has been introduced and solved polynomially in Section 3.2.3 that maximize the flow respecting the priority (cf. Problem 3.3, Algorithm 3,Theorem 3.4, and Corollary 3.3).

3.1 Abstract Flow

The abstract flow introduced in [36] generalizes the concept of paths by replacing the underlying network configuration. The maximum dynamic abstract flow problem and its solution procedure has been investigated by Kappmeier et al., [32]. The lexicographically maximum abstract flow problem has been investigated in [31]. This approach makes the use of so-called switching property that eliminates the crossing at intersections. Some of the important lane-based routing strategy for reducing the delays that reduce (or eliminate) crossing and merging conflicts at intersections have been studied in [37]. In a lane-based routing plan, selecting and turning options at intersections are restricted to improve traffic flow away from a hazardous area. Intersections with potentially significant delays can be temporarily transformed into an uninterrupted flow facility which is the most beneficial aspect of this routing.

Consider a multi-terminal evacuation network $N = (E, \Gamma, u, \theta, T, S, D)$ where E and Γ represent the sets of elements and paths, respectively. For every path $\gamma \in \Gamma$ there is a linear order $<_\gamma$ of elements and the set of such paths Γ satisfies the switching property in abstract network setting. A switching property requires that for each $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, there exist paths $\gamma_1 \times_e \gamma_2 \subseteq \{a \in \gamma_1 : a \leqslant_{\gamma_1} e\} \cup \{a \in \gamma_2 : a \geqslant_{\gamma_2} e\}$ and $\gamma_2 \times_e \gamma_1 \subseteq \{a \in \gamma_2 : a \leqslant_{\gamma_2} e\} \cup \{a \in \gamma_1 : a \geqslant_{\gamma_1} e\}$. The before and after parts from e of the path γ excluding e are denoted by $(\gamma, e) := \{p \in \gamma : p <_\gamma e\}$ and $(e, \gamma) := \{p \in \gamma : p >_\gamma e\}$, respectively. The generalized weighted maximum abstract flow problem formulated in [36] optimizes the value of Equation 3.1 satisfying Constraints 3.2 and 3.3.

$$\max \sum_{\gamma \in \Gamma} \omega(\gamma) g(\gamma), \tag{3.1}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} g(\gamma) \leq u_e, \; \forall \, e \in E, \tag{3.2}$$

$$g(\gamma) \geq 0, \; \forall \, \gamma \in \Gamma. \tag{3.3}$$

The weight function $\omega : \Gamma \longrightarrow \mathbb{R}_{>0}$ generalizes the maximum abstract flow problem by specifying the reward per unit of flow sent along each path. The choice of the weighted function ω is restricted to supermodular functions, i.e., $\omega(\gamma_1 \times_e \gamma_2) + \omega(\gamma_2 \times_e \gamma_1) \geq \omega(\gamma_1) + \omega(\gamma_2)$ for every $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, as the general problem is $\mathcal{NP}$-hard. The generalized minimum weight abstract cut is the dual of the generalized maximum weighted abstract flow problem, [36], which can be formulated as:

$$\min \sum_{e \in E} u_e x_e, \tag{3.4}$$

$$\sum_{e \in \gamma} x_e \geq \omega(\gamma),\ \forall\, \gamma \in \Gamma, \tag{3.5}$$

$$x_e \geq 0,\ \forall\, e \in E, \tag{3.6}$$

where a value x_e is assigned to every element $e \in E$ covering every path according to its weight.

The consideration of time factor transforms the abstract flow model into dynamic abstract flow model. The dynamic abstract flow problems can be transformed into static network by constructing the corresponding time-expanded networks. The time-expanded network can violate the switching property. For this, Kappmeier et al. [32] introduced the holdover of flow at intermediate nodes to construct an abstract time-expanded network. Suppose $\sigma : \Gamma \to \{1, 2, \ldots, T\}$ be waiting times for every elements of Γ. Then, the flow enters to $e \in \gamma$ at time $\sum_{q \in \gamma_{\to e}} (\sigma_q + \theta_q) + \theta_e$ as it traveling along γ waits σ_q time units before passing through e. The set $\gamma^\sigma := \{e^\kappa \in E^T | e \in \gamma, \sum_{q \in \gamma_{\to e}} (\sigma_q + \theta_q) + \theta_e = \kappa\}$ represents temporal paths with intermediate waiting and satisfy $q^\kappa <_{\gamma^\sigma} e^{\kappa'}$ if and only if $q <_\gamma e$. Such paths arrived within T is denoted and defined by $\Gamma_T^\sigma := \{\gamma^\sigma | \gamma \in \Gamma, \sigma \in \{1, 2, \ldots, T\}^\gamma, \sum_{e \in \gamma} \{\sigma_e + \tau_e\} \leq T\}$. The set (E_T, Γ_T^σ) represents the abstract time-expanded network (ATEN) of (E, Γ), where $E_T := \{(e, \tau) \mid e \in E, \tau \in \{1, 2, \ldots, T\}\}$.

Suppose $g_{dyn} : \Gamma_T^\sigma \to \mathbb{R}_{\geq 0}$ be the dynamic abstract flow. The feasible dynamic abstract flow satisfies the capacity of every element at every point of time. The maximum dynamic abstract flow problem maximizes the total flow value respecting the given restrictions, [32].

$$\max \sum_{\gamma_\tau \in \Gamma_T^\sigma} g_{dyn}(\gamma_\tau), \tag{3.7}$$

$$\sum_{\gamma_\tau \in \Gamma_T^\sigma : (e,\theta) \in \gamma_\tau} g_{dyn}(\gamma_\tau) \leq u_e,\ \forall\, e \in E,\ \theta \in \mathcal{T}, \tag{3.8}$$

$$g_{dyn}(\gamma_\tau) \geq 0,\ \forall\, \gamma_\tau \in \Gamma_T^\sigma. \tag{3.9}$$

The dynamic cut is $C_{dyn} := \{(e, \tau) \in E_T : \alpha(e) \leq \tau < \alpha(e) + \tilde{x}(e)\}$, where $\tilde{x}_e$ is static weighted abstract dual integral optimal solution with weight $\omega(\gamma)$ and

$$\alpha(e) := \min_{\gamma \in \Gamma} \sum_{\gamma \in (\Gamma, e)} (\theta(\gamma) + \tilde{x}(\gamma)).$$

Suppose $\gamma_1, \gamma_2 \in \Gamma$ are two paths with a common element $e \in \gamma_1 \cap \gamma_2$. The abstract network is terminal respecting, if all paths $\gamma \in \gamma_{1 \to e} \cup \gamma_{2e \to}$ satisfy first(γ) := first (γ_1) and last(γ):= last (γ_2). Let $d^+ : E \to Q_+$ be supplies for source elements and let $d^- : E \to Q_+$ be demands for sink elements. An abstract flow satisfies given supplies and demands if $\sum_{\gamma \in \Gamma : e = \text{first}(\gamma)} g_\gamma = d_e^+$ and $\sum_{\gamma \in \Gamma : e = \text{ last}(\gamma)} g_\gamma = d_e^-$ holds for source and sink elements, respectively. It is possible that an element is a source and a sink both.

Let g be an abstract flow in (E_T, Γ_T^σ). For any $\tau \in \{1, 2, \ldots, T\}$, consider the set of temporal paths $\Gamma_\tau^\sigma := \{\gamma \in \Gamma | \sum_{e \in \gamma} \{\sigma_e + \theta_e\} \leq \tau\}$. Define the flow with value $|g_{dyn}|_\tau := \sum_{\gamma \in \Gamma_\tau} g_\gamma$ that arrives until time τ. An earliest arrival abstract flow problem is to find a maximum flow such that it holds $|g_{dyn}|_\tau \geq |g_{dyn}^\tau|$ for all τ, where $|g_{dyn}^\tau|$ is the value of a maximum dynamic abstract flow with time horizon τ.

Let $|g|_s^+ := \sum_{\gamma \in E_s^+} g_\gamma$, and $|g|_t^- := \sum_{\gamma \in E_t^-} g_\gamma$ be the out going and incoming abstract flow from the source s and to the sink t, respectively. Let g^1 and g^2 be maximum abstract flows, and let $s_1, s_2, \ldots, s_k$ and $t_1, t_2, \ldots, t_k$ be the orders of sources and sinks, respectively. We say that g^2 is lexicographically smaller than g^1, denoted by $g^1 \geq_L g^2$, if there exists either an $l \in \{0, 1, \ldots, k-1\}$ such that $|g^1|_{s_{l+1}}^+ > |g^2|_{s_{l+1}}^+$ and $|g^1|_{s_i}^+ = |g^2|_{s_i}^+$ for $i = 1, 2, \ldots, l$, or all $|g^1|_{s_i}^+ = |g^2|_{s_i}^+$ for $i = 1, \ldots, k$. Similarly, $g^1 \geq_L g^2$ if either $|g^1|_{t_{l+1}}^- > |g^2|_{t_{l+1}}^-$ and $|g^1|_{t_i}^- = |g^2|_{t_i}^-$ for some $l \in \{0, 1, \ldots, k-1\}$ and $i = 1, 2, \ldots, l$ or $|g^1|_{t_i}^- = |g^2|_{t_i}^-$ for all $i = 1, \ldots, k$. The lexicographically maximum abstract flow (LMAF) g^* is a maximum abstract flow (MAF) respecting the terminal orders, i.e. $g^* \geq_L g$ for all abstract flows g.

A sequence of terminals is compatible if the terminal elements respect their rank for more than one terminals of the same type appeared on a path. For sources $s_1, s_2, \ldots, s_k$, it holds that $\gamma \in \Gamma, s_i \neq s_j \in \gamma : j < i \Rightarrow s_i \leq_\gamma s_j$. But a sequence of sinks $t_1, t_2, \ldots, t_k$ has to assure $\gamma \in \Gamma, t_i \neq t_j \in \gamma : i < j \Rightarrow t_i \leq_\gamma t_j$. For a given compatible sequence of sources and sinks, we define abstract networks (E, Γ_s^i) with increasing subsets of paths $\Gamma_s^i \subset \Gamma$ for $i = 1, 2, \ldots, k$, where $\Gamma_s^i = \Gamma_s^{i-1} \cup \{\gamma \in \Gamma | s_i = \text{first}(\gamma)\}$ with $\Gamma_s^0 = \emptyset$. Similarly, we define (E, Γ_t^i) with paths $\Gamma_t^i \subset \Gamma$, where $\Gamma_t^i = \Gamma_t^{i-1} \cup \{\gamma \in \Gamma | t_i = \text{last}(\gamma)\}$ with $\Gamma_t^0 = \emptyset$. Each of (E, Γ_s^i) and (E, Γ_t^i) contains the paths starting and ending in the first i sources and sinks, respectively.

The central idea behind the contraflow technique is to improve the outbound capacity by adopting the arc or path reversals toward the safer places keeping the same travel time in the evacuation network. As a result the flow value is increased, evacuation time is decreased and traffic flow is made smooth. Let $\Gamma = \{\overrightarrow{\gamma}, \overleftarrow{\gamma}\}$ be the set of all paths in abstract evacuation network with capacities $u(\overrightarrow{\gamma}) = \min\{u_e : e \in \overrightarrow{\gamma}\}$ and $u(\overleftarrow{\gamma}) = \min\{u_e : e \in \overleftarrow{\gamma}\}$. We define the undirected auxiliary network $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, s, t)$ by adding the capacities on the corresponding two-way paths and keeping the transit time (if any) fixed. The set of elements and paths are denoted by $\tilde{E}$ and $\tilde{\Gamma}$, where $\tilde{e} \in \tilde{E}$ and $\tilde{\gamma} \in \tilde{\Gamma}$. Then, the capacity function is defined as $\tilde{u}(\tilde{\gamma}) = \min\{\tilde{u}_{\tilde{e}} : \tilde{e} \in \tilde{\gamma}\}$ while the travel times (if any) on paths remains the same.

By construction of auxiliary network with path reversal, $\tilde{\Gamma}$ satisfies the switching property and the order of elements holds for each $\tilde{\gamma} \in \tilde{\Gamma}$. This implies that the auxiliary network of the abstract evacuation network is also an abstract network.

3.2 Abstract Contraflow Problems

In this section, we define abstract contraflow problems and present efficient solution procedures to solve them. This approach uses path reversals in abstract network at time zero without any switching costs.

Example 3.1. Let $E = \{s, a, b, c, d, e, t\}$, $\Gamma = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overrightarrow{\gamma_3}, \overrightarrow{\gamma_4}, \overrightarrow{\gamma_5}, \overrightarrow{\gamma_6}, \overleftarrow{\gamma_1}, \overleftarrow{\gamma_2}, \overleftarrow{\gamma_3}, \overleftarrow{\gamma_4}, \overleftarrow{\gamma_5}, \overleftarrow{\gamma_6}\}$ with $\overrightarrow{\gamma_1} = \{s, a, c, t\}$, $\overrightarrow{\gamma_2} = \{s, b, d, t\}$, $\overrightarrow{\gamma_3} = \{s, a, e, d, t\}$, $\overrightarrow{\gamma_4} = \{s, b, e, c, t\}$, $\overrightarrow{\gamma_5} = \{s, a, e, c, t\}$, $\overrightarrow{\gamma_6} = \{s, b, e, d, t\}$, $\overleftarrow{\gamma_1} = \{t, c, a, s\}$, $\overleftarrow{\gamma_2} = \{t, d, b, s\}$, $\overleftarrow{\gamma_3} = \{t, d, e, a, s\}$, $\overleftarrow{\gamma_4} = \{t, c, e, b, s\}$, $\overleftarrow{\gamma_5} = \{t, c, e, a, s\}$ and $\overleftarrow{\gamma_6} = \{t, d, e, b, s\}$. We forget the direction of paths and reformulate it by adding the capacities of paths between the terminals. Set of paths in abstract auxiliary network is $\tilde{\Gamma} = \{\tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{\gamma}_3, \tilde{\gamma}_4, \tilde{\gamma}_5, \tilde{\gamma}_6\}$ (cf. Figure 3(ii)).

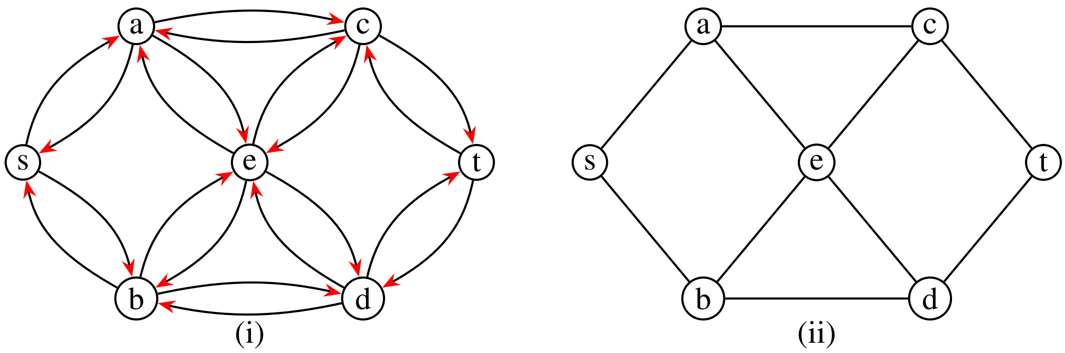


Figure 3: Abstract evacuation and auxiliary networks, respectively.

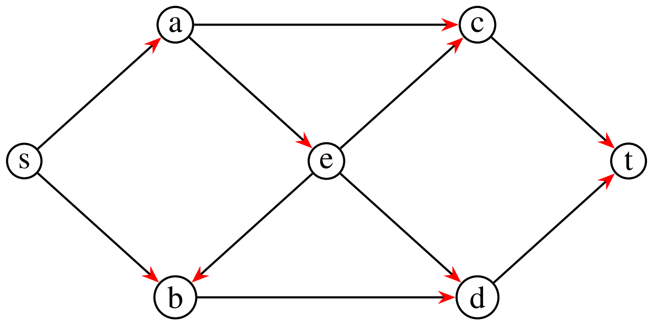


Figure 4: Abstract network after contraflow reconfiguration in Figure 3(i).

3.2.1 Maximum abstract contraflow

We define the maximum abstract contraflow (MACF) by integrating the contraflow model introduced in [15, 16] and the MAF problem solved in [72]. With simple extension, we also give an efficient algorithm for solving Problem 3.1.

Problem 3.1. Given an abstract network $N = (E, \Gamma, u, s, t)$, the MACF problem is to find the MAF with path reversals allowed initially.

Theorem 3.1. Let $\tilde{u} \in Z^+$ be the capacities and $\omega_{\tilde{\gamma}}$ be the supermodular weights on paths $\tilde{\gamma} \in \tilde{\Gamma}$. Then, the weighted abstract flow and weighted abstract cut problems in auxiliary abstract network (AAN) have totally dual integer optimum solutions.

To get solution for MACF, we combine the concepts of path reversal criterion presented in [14] and MAF algorithm provided in [72].

Lemma 3.1. The MACF does not decrease the flow value after contraflow configuration.

Proof. By definition, $C \subseteq E$ is the collection of disconnecting and saturated elements such that every path connects source and sink, and contains only one element from $\tilde{\gamma}$. By construction, $\tilde{u}(\tilde{C}) = \sum_{e \in \tilde{C}} \tilde{u}_e \geq \sum_{e \in C} u_e$, where $\tilde{C} \subseteq \tilde{E}$ represents the cut in auxiliary network. Following Theorem 3.1, we have

$$\begin{aligned} val_{max}(\tilde{g}) &= max \sum_{\tilde{\gamma} \in \tilde{\Gamma}} g(\tilde{\gamma}) \\ &= min\{\sum_{\tilde{e} \in \tilde{C}} \tilde{u}_{\tilde{e}} : \tilde{C} \subseteq \tilde{E}\} \geq min\{\sum_{e \in C} u_e : C \subseteq E\} \\ &= max \sum_{\gamma \in \Gamma} g(\gamma) = val_{max}(g). \end{aligned}$$

Thus the claim follows. □

Lemma 3.2. [14] The abstract contraflow doubles the flow value after contraflow configuration whenever each element in a minimum abstract cut has symmetric capacity.

Algorithm 1: Maximum Abstract Contraflow

Input : Two-terminal path reversible abstract network $N = (E, \Gamma, u, s, t)$.

1. Construct, $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{s}, \tilde{t})$ with new capacity $\tilde{u}(\tilde{\gamma}) = u(\overrightarrow{\gamma}) + u(\overleftarrow{\gamma})$.
2. Solve the maximum abstract network flow problem in $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{s}, \tilde{t})$ using [72] as follows:
   (a) Initialize $\tilde{g} = \tilde{g}_0$, if an initial solution is given, otherwise initialize as the zero flow.
   (b) While $\tilde{g}$ is not optimal:
      i. Compute an augmenting structure. If no such structure exists, return $\tilde{g}$.
      ii. Determine $\delta \in \tilde{N}$ so that all paths in augmenting structure can be augmented by δ.
      iii. For each path $\tilde{\gamma}^+$ in augmenting structure, set $\tilde{g}_{\tilde{\gamma}^+} = \tilde{g}_{\tilde{\gamma}^+} + \delta$.
      iv. For each path $\tilde{\gamma}^-$ in augmenting structure, set $\tilde{g}_{\tilde{\gamma}^-} = \tilde{g}_{\tilde{\gamma}^-} - \delta$.
3. A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $u(\overrightarrow{\gamma})$ or there is a non-negative flow along the path $\overrightarrow{\gamma} \notin \Gamma$.

Output: Maximum abstract contraflow value in $N = (E, \Gamma, u, s, t)$.

Theorem 3.2. Algorithm 1 solves Problem 3.1 optimally.

Proof. Steps 1 and 2 are feasible by definition. Step 3 is well defined; i.e. not both paths $\overrightarrow{\gamma}$ and $\overleftarrow{\gamma}$ have to be switched at a time, this is ensured by the solution of the abstract flow in auxiliary network, [36]. Switching property cancels cycle flows so that there is a flow along $\overrightarrow{\gamma}$ or $\overleftarrow{\gamma}$ but not in both directions at the same time. Hence, the resulting flow from Step 3 is a feasible flow with path reversals in N.

Weights are supermodular and abstract path system satisfies switching property so that the MAF is totally dual integral with minimum cut, [36]. By equivalence of optimization and separation problems of [153], the oracle solves the separation problem for weighted abstract flow problem. The abstract maximum flow algorithm maintains a candidate set for an abstract minimum cut as solution of dual problem. The algorithm then calls oracle to verify the dual feasible solution, infact this is the case, the primal solution is a MAF, [72]. Otherwise, the oracle returns a violating path. The returned violated paths are then combined to an augmenting structure which allows to improve the flow value. In fact, any optimal solution to the maximum flow problem with path reversals on N is also a feasible solution to the maximum flow problem on $\tilde{N}$. As the amount of flow sent from s to t in Steps 2 is not changed in Step 3, the resulting flow is an optimal. □

Corollary 3.1. [14] Algorithm 1 computes MACF solution in polynomial-time.

Proof. The direction of paths can be reversed using Step 3 in linear time. Construction of AAN takes linear time. Thus, the complexity of Algorithm 1 depends on the complexity of Step 2 that takes $\mathcal{O}(|\tilde{E}|logU)$ time in $\tilde{N}$, where U is the maximum capacity of any path $\tilde{\gamma} \in \tilde{\Gamma}$. □

Example 3.2. Consider the abstract network of Figure 5 (i), where 4 units flow through $(s; a; t)$, 2 units flow through $(s; b; t)$ and 1 unit flow through $(s; a; b; t)$ can be send from source to sink. Here, the MAF is 7 in the network. But it is 13 for the network of Figure 6 after contraflow configuration with following path flows: 6 units through $(s; a; t)$, 6 units through $(s; b; t)$, and 1 units through $(s; a; b; t)$.

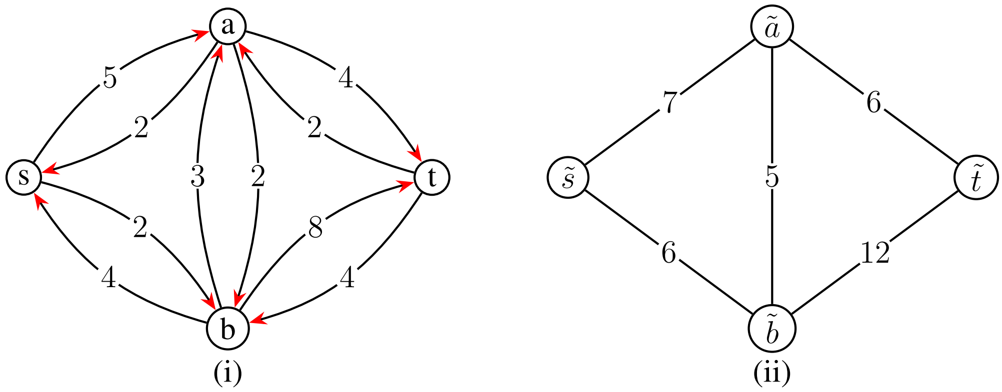


Figure 5: Abstract evacuation and auxiliary network, respectively.

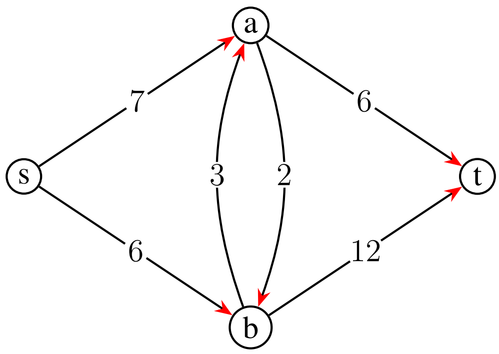


Figure 6: Reconfigured abstract network of Figure 5(i).

3.2.2 Maximum dynamic abstract contraflow

The MDAF problem has been introduced and solved in [32]. The maximum dynamic contraflow problem with arc reversal capability and its strongly polynomial-time solution procedure are presented by [16]. We introduce and solve the maximum dynamic abstract contraflow (MDACF) with path reversal capability (cf. Problem 3.2).

Problem 3.2. Given an abstract network $N = (E, \Gamma, u, \theta, T, s, t)$, the MDACF problem is to find the MDAF with path reversal capability at time zero.

The abstract path system is allowed to be asymmetric with respect to the path capacities but the transit times are symmetric. In auxiliary network only capacities of the paths change but the transit times remain the same. As the abstract cut C_{dyn} contains an element of every temporally repeated paths, the capacity constraints are satisfied at each point of time. Thus for a subset $C_{dyn} \subseteq E_T$, the set $\gamma_\tau \cap C_{dyn}$ is non-empty to each $\gamma_\tau \in \Gamma_T$, [32]. This implies that $\sum_{\gamma_\tau \in \Gamma_T} g_{dyn}(\gamma_\tau) \leq \sum_{(e,\,\tau) \in C_{dyn}} u_e$. The number of paths created by applying the time expansion is linear in T and thus exponential in the size of input. We combine the concept of path reversal capability of [14] and solution method of MDAF problem of [32]. Algorithm 2 works in ATEN of an AAN that consists of a (static) abstract network. For each interval, a copy of the element set $\tilde{E}$, the element set $\tilde{E}_T := \tilde{E} \times \mathcal{T}$ will be constructed in ATEN is constructed by $\tilde{E}_T := \{(\tilde{e}, \tau) | \tilde{e} \in \tilde{E}, \tau \in \{1, 2, \ldots, T\}\}$.

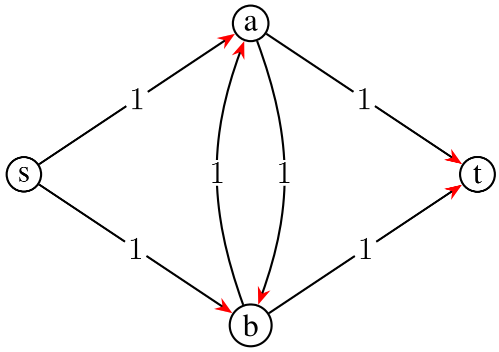


Figure 7: Dynamic abstract network.

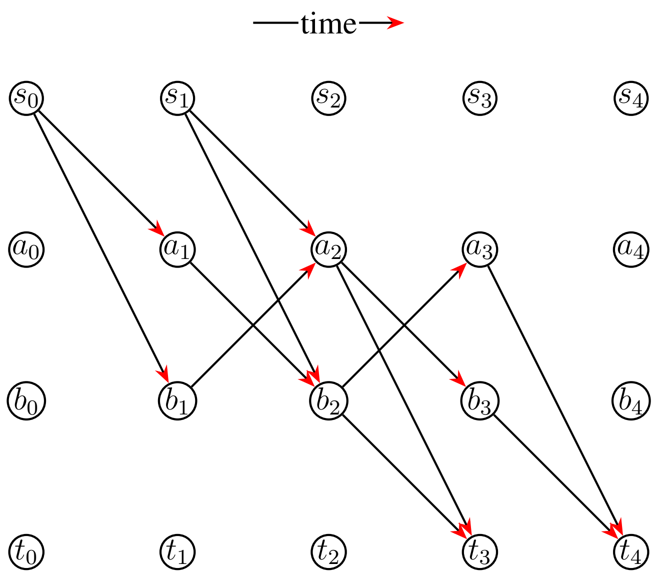


Figure 8: Time expansion of Figure 7.

Example 3.3. The ATEN of an AAN can destroy the switching property. Let $(\tilde{E}, \tilde{\Gamma})$ be an auxiliary network with $\tilde{\Gamma} = \{\gamma^1, \gamma^2, \gamma^3, \gamma^4\}$ and $\tilde{E} = \{s, a, b, t\}$, where $\gamma^1 = (s;\ a;\ b;\ t)$, $\gamma^2 = (s;\ b;\ a;\ t)$, $\gamma^3 = (s;\ a;\ t)$, $\gamma^4 = (s;\ b;\ t)$ together with their reversals. The set $\tilde{\Gamma}$ satisfies the switching and order properties. We have $\Gamma_T = \{\gamma_0^1, \gamma_2^1, \gamma_0^2, \gamma_1^2, \gamma_0^3, \gamma_1^3, \gamma_2^3, \gamma_0^4, \gamma_1^4, \gamma_2^4\}$, for $T = 4$, with $\gamma_0^1 = \{(s, 0); (a, 1,); (b, 2); (t, 3)\}$

and $\gamma_1^2 = \{(s,1);(b,2);(a,3);(t,4)\}$. But time expansion of given network does not contain $\gamma_0^1 \times_{(b,2)} \gamma_1^2$ (cf. Figure 8).

As in the abstract dynamic network, we can construct time-expanded abstract network $(\tilde{E}_T, \tilde{\Gamma}_T^\sigma)$ of the auxiliary abstract dynamic network $(\tilde{E}, \tilde{\Gamma})$ with time horizon T. In this case, internal waiting does not make difference in optimality, [32].

Lemma 3.3. The MDACF in $N = (E, \Gamma, u, \theta, s, t, T)$ is not more than the optimal flow in the MACF problem for the corresponding time-expanded network.

Proof. The maximum amount of flow in the single source and single sink dynamic contraflow problem is less than the optimal flow in the maximum contraflow problem for the corresponding time-expanded network, [16]. The ATEN can be constructed by allowing intermediate waiting that does not make difference in optimality of MDAF, [32]. Then we can conclude that every feasible flow to the MDACF problem has an equivalent feasible flow to the maximum contraflow problem of the time-expanded graph. Hence, the claim follows. □

Lemma 3.4. The dynamic abstract contraflow does not decrease the flow value after contraflow.

Proof. By definition of minimum dynamic cut, $C_{dyn} \subseteq E_T$ is the collection of disconnecting and saturated elements with each path connecting source and sink in ATEN. The element set $\tilde{E}$ is copied for each point of time to construct the element set $\tilde{E}_T := \tilde{E} \times \mathcal{T}$ in ATEN. The network $(\tilde{E}, \tilde{\Gamma})$ can destroy switching property. To construct abstract network, waiting is allowed in intermediate nodes but there is not any difference in optimality, [32]. Every path passes through the cut set exactly onces on every time interval, i.e., $\tilde{\Gamma}^\sigma \cap \tilde{C}_{dyn} \neq \emptyset$, [32]. The $\tilde{C}_{dyn}$ contains only one element from $\tilde{\gamma}$. The minimum cut capacity becomes $\tilde{b}(C_{dyn}^*) = \sum_{(\tilde{e},\theta)\in C^*} \tilde{b}_e\}$, where $\tilde{C}_{dyn}^*$ is the minimum cut. If each element $(e,\theta) \in C_{dyn}^*$ has capacity in both directions, the contraflow reconfiguration of the abstract network increases the capacity of cut but the capacity of minimum cut will not decrease even if there is not capacity in both directions. Then

$$\tilde{u}(\tilde{C}^*_{dyn}) = \sum_{(\tilde{e},\theta)\in\tilde{C}^*_{dyn}} \tilde{b}_{(\tilde{e},\tau)} \geq \sum_{(e,\theta)\in C^*_{dyn}} u_{(e,\theta)}.$$

Every MDAF is equal to the minimum dynamic abstract cut in auxiliary network, [32]. Then,

$$\begin{aligned} val_{max}(\tilde{g}_{dyn}) &= max \sum_{\tilde{\gamma}_\tau \in \tilde{\Gamma}^\sigma_T} \tilde{g}(\tilde{\gamma}_\tau) \\ &= min\{\sum_{\tilde{e}\in\tilde{C}} \tilde{u}_{\tilde{e}} : \tilde{C}_{dyn} \subseteq \tilde{E}_T\} \\ &\geq min\{\sum_{e\in C} u_e : C_{dyn} \subseteq E_T\} \\ &= max \sum_{\gamma_\tau \in \Gamma^\sigma_T} g(\gamma_\tau) \\ &= val_{max}(g_{dyn}). \end{aligned}$$

The MDAF in AAN $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{s}, \tilde{t})$ is a feasible dynamic abstract contraflow in original abstract network $N = (E, \Gamma u, \theta, T, s, t)$ with path reversal capability. Hence, the claim follows. □

Lemma 3.5. The contraflow reconfiguration of an abstract contraflow network increases the flow value two times if each element in a minimum abstract dynamic cut has symmetric capacity.

Algorithm 2: Maximum Dynamic Abstract Contraflow

Input : Path reversible abstract dynamic network $N = (E, \Gamma, u, \theta, T, s, t)$.

1. Construct, $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{s}, \tilde{t})$ with new capacity and transit time functions $\tilde{u}(\tilde{\gamma}) = u(\overrightarrow{\gamma}) + u(\overleftarrow{\gamma})$ and $\tilde{\theta}(\tilde{\gamma}) = \sum_{\tilde{e}\in\tilde{\gamma}} \tilde{\theta}(\tilde{e})$.

2. Generate a temporally repeated dynamic flows $\tilde{N}$ with capacity $\tilde{u}(\tilde{\gamma})$ and transit time $\tilde{\theta}(\tilde{\gamma})$, [32].

3. A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $u(\overrightarrow{\gamma})$ or there is a non-negative flow along path $\overrightarrow{\gamma} \notin \Gamma$.

Output: Maximum abstract contraflow in $N = (E, \Gamma, u, \theta, T, s, t)$.

Theorem 3.3. Algorithm 2 solves Problem 3.2 optimally.

Proof. Steps 1 and 2 are feasible. Constructed flows from Step 2 are temporally repeated, there is only a flow in one direction of two elements at the same time as well as at different time periods, this ensures that the flow is less than the reversed capacities on all the paths at all time units and also ensures the feasibility of Step 3. Since every feasible flow of the DAF problem in $\tilde{N}$ is feasible to the MDACF problem in N, the MDACF in N is not more than the MDAF in $\tilde{N}$. The MACF in ATEN is not less than the MDACF in N, Lemma 3.3. Hence, the optimal dynamic contraflow in N is not greater than the MACF in N_T. The MACF problem in N_T is equivalent to the maximum flow problem in $\tilde{N}_T$, Theorem 3.2. Thus, the MACF in N_T is equal to the maximum abstract flow (MAF) in $\tilde{N}_T$. The temporally repeated abstract flow g^*_{dyn} is a MDAF, and C_{dyn} is a minimum abstract cut over time whose capacity is equal to the flow value, [32]. Thus the maximum flow can be obtained by a temporally repeating a path flow of a static graph $\tilde{N}$. Hence the MAF in $\tilde{N}_T$ is equal to the MDAF in $\tilde{N}$. □

Corollary 3.2. Algorithm 2 computes the MDACF in polynomial time.

Proof. Steps 1 and Step 3 can be completed in linear time, so that the complexity is dominated by Step 2 that computes the MDAF problem in AAN. Step 2 can be computed in polynomial time, [32]. □

Example 3.4. Consider the symmetric transit time $\theta_e = 1$ for each $e \in E$ in Figure 5 (i). Using the algorithm, 6 units flow at time 4 through the paths $(s; b; t)$ and $(s; a; t)$, 7 units flow at time 5 through the paths $(s; b; t)$, $(s; a; t)$, and $(s; a; b; t)$ can be send to the sink. By the same idea, dynamic flows can be calculated in the network after contraflow configuration. The MDAF at time $T = 5$ are 13 and 25 before and after contraflow configuration, respectively, showing the significant increment with contraflow.

3.2.3 Lexicographically maximum abstract contraflow

If we have a given rank on the terminals with priorities, the flow is compared by its value in their rank ordering, referred to as lexicographically maximum flow. An existence and a polynomial solution of LMAF problem have been presented in [31]. In his model, the order of terminals has to fulfill compatible property if more than one terminal node is contained in a path.

Problem 3.3. Let $N = (E, \Gamma, u, S, D)$ be a multi-terminal abstract network, where S and D represent compatible set of of sources and sinks, respectively. The lexicographically maximum abstract contraflow (LMACF) problem is to find a LMAF where paths can be reversed without any cost.

Based on the LMACF algorithm, [31] and the lexicographically maximum contraflow algorithm [19], we propose LMACF Algorithm 3, which solves the Problem 3.3 in polynomial-time.

Algorithm 3: Lexicographically Maximum Abstract Contraflow

Input : Abstract network $N = (E, \Gamma, u, S, D)$ with a compatible sequence of sources $s_1, s_2, \ldots, s_k$ or sinks $t_1, t_2, \ldots, t_k$ in S and D, respectively, and weight $\omega \equiv 1$).

1. Construct, $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{S}, \tilde{D})$ with capacity $\tilde{u}(\tilde{\gamma}) = u(\overrightarrow{\gamma}) + u(\overleftarrow{\gamma})$.

2. Find solution in auxiliary network using Abstract Lexicographically Maximum Flow Algorithm,[31]:

   (a) Set $i = 0$ and initialize $\tilde{g}^0 = 0$ as the zero flow on all paths.

   (b) Set $i = i + 1$ and define the abstract networks $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$, (or $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{t}})$ w. r. t. the sinks).

   (c) Compute a flow $\tilde{g}^i$ using Step 2 of Algorithm 1 in $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$ starting with solution $\tilde{g}^{i-1}$.

   (d) If $i = k$ return $\tilde{g}^k$, otherwise continue with Step 2b.

3. A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $u(\overrightarrow{\gamma})$ or there is a non-negative flow along path $\overrightarrow{\gamma} \notin \Gamma$.

Output: Lexicographically maximum abstract contraflow in $N = (E, \Gamma, u, S, D)$.

Lemma 3.6. Let $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_k$ be a compatible sequence of sources and let $\tilde{g}^i$ be a MAF in the auxiliary network $(\tilde{A}, \tilde{\Gamma}^i_{\tilde{s}})$. If LMAF algorithm is executed in $(\tilde{A}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$ with initial flow $\tilde{g}^i$ as input, the flow value $|\tilde{g}|^+_{\tilde{s}_j}$ does not decrease for $j = 1, \ldots, i$ during the execution. Moreover, the same is true for sinks.

Theorem 3.4. Algorithm 3 solves Problem 3.3 in $N = (E, \Gamma, u, S, D)$ optimally.

Proof. The feasibility can be proved as in Theorem 3.2. The path capacity is increased by adding the capacity of paths and either direction of them is allowed in auxiliary network with unaltered priority ordering. Step 2 works for source and sink element sequences in auxiliary network. We prove the optimality by induction on i. The first iteration computes a MAF $\tilde{g}^1$ in $(\tilde{E}, \tilde{\Gamma}^1_{\tilde{s}})$ with single source. Let $\tilde{g}^i$ be a LMAF in

$(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$. By Lemma 3.6, flow $\tilde{g}^{i+1}$ does not reduce the inflow of source element $\tilde{s}_j$ for some $j \in \{1, 2, \ldots, i\}$ and the flow is maximum. Assume now that $\tilde{g}^{i+1}$ is not a LMAF in the abstract network $(\tilde{E}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$. Then there is a flow $\tilde{g}^{'}$ that sends more flow out of the source $\tilde{s}_j$ for some $j \in \{1, 2, \ldots, i\}$. Define the restricted $\tilde{g}^r$ by setting $\tilde{g}^r$=$\tilde{g}^{'}$ for each path $\gamma \in \Gamma^i$. The outflow of source element $\tilde{s}_j$ is the same for $\tilde{g}^{'}$ and $\tilde{g}^r$, and $\tilde{g}^r$ is a feasible abstract flow in $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$ that sends more flow out of $\tilde{s}_j$ than does $\tilde{g}^i$ which contradicts that $\tilde{g}^i$ is a lexicographically maximum. Hence, $\tilde{g}^{i+1}$ is a LMAF in $(\tilde{E}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$. Any optimal solution to the LMAF problem with path reversal in graph (E, Γ) is also a feasible solution to the maximum flow problem in $\tilde{N}$. As the amount of flow sent from $\tilde{S}$ to $\tilde{D}$ in Step 2 is not changed in Step 3, the resulting flow is optimal for the Problem 3.3. □

Corollary 3.3. Algorithm 3 solves the LMACF problem in polynomial-time complexity.

Proof. The construction of auxiliary network and Step 3 can be completed in linear time. As $N = (E, \Gamma, u, S, D)$ is an abstract network with a compatible sequence of sources, the complexity depends upon the running time of Step 2. Moreover, a separation oracle is given in the problem. The LMAF can be computed in polynomial-time which is dominated by $|\tilde{E}|$, [31]. □

3.2.4 Earliest arrival abstract contraflow

The existence of earliest arrival abstract flow (EAAF) described by [31] generalizes the earliest arrival flow which maximizes the DAF at each possible time point.

Problem 3.4. Let $N = (E, \Gamma, u, \theta, T, s, t)$ be a two-terminal abstract dynamic network. The earliest arrival abstract contraflow (EAACF) problem is to find the EAAF with path reversal capability at time zero.

The earliest arrival flow problem is not in the class $\mathcal{P}$. Based on the earlier results of earliest arrival contraflow problem with arc reversals in [19] and EAAF in [31], we introduce EAACF problem and propose Algorithm 4 to solve it. Minieka, [73] proved that using the successive shortest path algorithm in the original network and sending flow along the generalized temporally repeated paths leads to an earliest arrival flow. This approach does not require waiting in intermediate nodes and sends flow only along temporal copies of the original paths. A similar algorithm to the successive shortest path algorithm is in [154]. Their algorithm computes a maximum weighted abstract flow by using augmenting structure of decreasing total reward, where the shortest path has the most reward.

Algorithm 4: Earliest Arrival Abstract Contraflow

Input : Abstract network $N = (E, \Gamma, u, \theta, T, s, t)$, where paths can be reverse without any cost.

1. Construct, $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{s}, \tilde{t})$ with new capacity and transit functions $\tilde{u}(\tilde{\gamma}) = u(\overrightarrow{\gamma}) + u(\overleftarrow{\gamma}),$ and $\tilde{\theta}(\tilde{\gamma}) = \sum_{\tilde{e}\in\tilde{\gamma}} \tilde{\theta}(\tilde{e})$, respectively.

2. Solve the problem in the auxiliary network using Step 2 of Algorithm 3 in corresponding $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ of $\tilde{N}$.

3. A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $u(\overrightarrow{\gamma})$ or there is a non-negative flow along path $\overrightarrow{\gamma} \notin \Gamma$.

Output: Earliest arrival abstract contraflow in $N = (E, \Gamma, u, \theta, T, s, t)$.

Theorem 3.5. Algorithm 4 solves Problem 3.4 optimally.

Proof. The feasibility can be ensured as in Theorem 3.2. Let $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ be the corresponding increasing ATEN of $\tilde{N}$ with intermediate waiting, [32], where T be a significantly large time horizon. Step 2 is executed in $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ that avoids reduction of flow sent in earlier time steps by removing backward paths in the residual network, Lemma 3.6. For an abstract network $(\tilde{E}, \tilde{\Gamma}, T)$, there exists an EAAF flow $\tilde{g}_{dyn}$ in $(\tilde{E}, \tilde{\Gamma}, T)$. The resulting flow $\tilde{g}_{dyn}$ from Step 2 signifies that the obtained flow is optimal in $(\tilde{E}, \tilde{\Gamma}, T)$. Any optimal solution to the earliest arrival flow problem with path reversal in network N is also a feasible solution to the EAAF problem in the auxiliary network $\tilde{N}$. As the amount of flow sent from s to t in Step 2 is not changed in Step 3, the resulting flow is an optimal solution to the EAACF problem in N. □

Example 3.5. Consider the network of Example 3.4. In this example, MDACF satisfies the earliest arrival flow property. Here, the EAAF values before contraflow configuration at time $T = 4$ and 5 are 6 and 13, respectively. Also, the EAACF values at these times are 12 and 25, respectively.

3.2.5 Abstract earliest arrival transshipment

We study abstract earliest arrival contraflow problem on multi-terminal network. Clearly, the abstract earliest arrival contraflow cannot be obtained for multiple sinks. But the flow value can be approximated by adding supplies and demands for source and sink elements, respectively. Based on the results of [20, 31] on contraflow and abstract flow, we introduce abstract earliest arrival contraflow Problem 3.5 and propose an approximation Algorithm 5 to solve it. Recall that, the abstract contraflow transshipment having the earliest arrival property does not exist.

Problem 3.5. Let $N = (E, \Gamma, u, \theta, T, S, D)$ be an abstract network. The multi-terminal abstract earliest arrival contraflow problem is to find an abstract earliest arrival flow with path reversals capability that satisfies the supplies and demands at terminals.

Example 3.6. Consider an abstract network with elements $E = \{s, a, b, c, t_1, t_2\}$ and paths $\Gamma = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overleftarrow{\gamma_1}, \overleftarrow{\gamma_2}\}$, $\overrightarrow{\gamma_1} = \{s; a; c; t_1\}$, $\overrightarrow{\gamma_2} = \{s; a; b; t_2\}$, $\overleftarrow{\gamma_1} = \{t_1; c; a; s\}$, and $\overleftarrow{\gamma_2} = \{t_2; b; a; s\}$. Suppose $d_s^+ = 12, d_{t_1}^- = 4, d_{t_2}^- = 8, d_a = 0, d_b = 0,$ and $d_c = 0$ be the balance vectors. Construct auxiliary network with $\tilde{\Gamma} = \{\tilde{\gamma_1}, \tilde{\gamma_2}\}$, $\tilde{\gamma_1} = \{\tilde{s}; \tilde{a}; \tilde{c}; \tilde{t_1}\}$, and $\tilde{\gamma_2} = \{\tilde{s}; \tilde{a}; \tilde{b}; \tilde{t_2}\}$ which is shown in Figure 9.

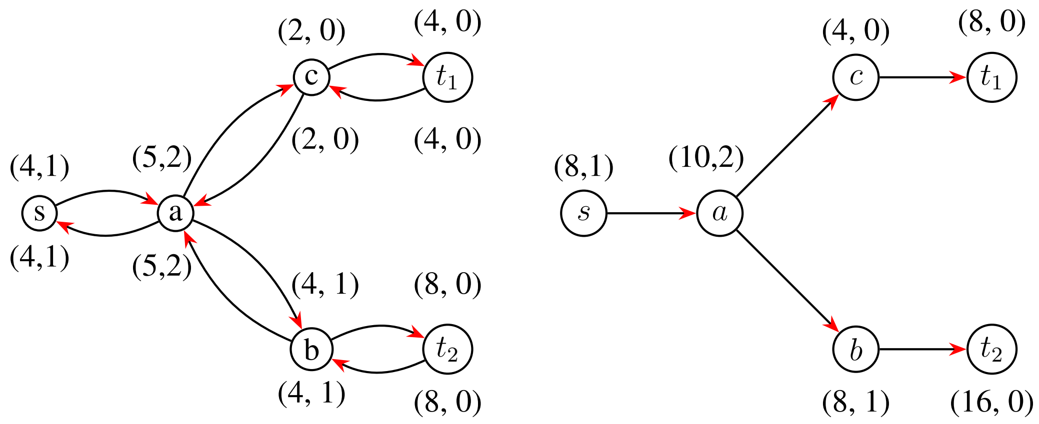


Figure 9: Multi-terminal abstract networks before and after reconfiguration, respectively.

Besides the given feasible transshipment at time 5 there is a feasible transshipment which sends 4 units of flow on path $\tilde{\gamma}_1$ at time 4, 4 units of flow on path $\tilde{\gamma}_2$ at time 5 and another 4 units of flow on path $\tilde{\gamma}_2$ at time 6. Here, 4 units of flow arrive at time 4 because $\tilde{\gamma}_1$ has 3 transit time, another 4 units arrive at time 5 through path $\tilde{\gamma}_2$ and next 4 units arrive at time 6 through path $\tilde{\gamma}_2$. This solution sends 4 units of flows earlier, but needs more time to send the last 4 units of flows. Hence, no earliest arrival transshipment contraflow exists in the network and the new solution is a 2-value-approximate flow.

The proposed algorithm works on extended path system having two restrictions such that every sub-paths should preserve orders and no flow can exceed the demands and supplies, [31], as constructed below. Let (E_T, Γ_T^σ) be the abstract time expanded network for some time horizon T. We introduce additional super source elements s^*, super sink elements t^* and counting elements t_c. The time-expanded ground set is defined as: $E_T^* := E_T \cup \{s_e^* : s \in S\} \cup \{t_e^* : t \in D\} \cup \{e_c^\tau : e \in D, \tau \in \{1, 2, \ldots, T\}\}$. We extend each original temporal path $\gamma^\sigma = (e_1, e_2, \ldots, e_n) \in \Gamma_T^\sigma$ by the corresponding super terminals and a counting element. Let $s = e_1 = \text{first}(\gamma)$ and $t^\tau = e_n = \text{last}(\gamma)$ be the first and last element of γ^σ, respectively. The extended path $\gamma^{*\sigma} := (s^*, e_1, e_2, \ldots, e_n, t_c^\tau, t^*)$ contains three more elements, the super source as new first element, the super sink as last element and the counting element left of the super sink. The set of extended paths $\Gamma_T^{*\sigma}$ consists of all extended paths $\gamma^{*\sigma}$ with internal waiting at intermediate elements.

Example 3.7. Figure 10 is the extended time-expanded network derived from Figure 8.

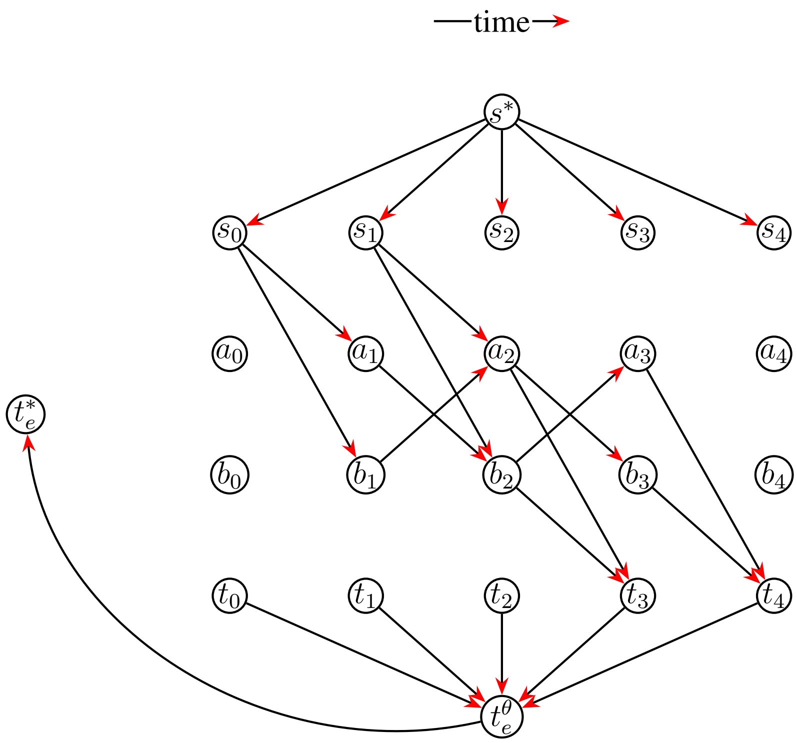


Figure 10: Extended time-expanded network of Figure 8.

The primal dual pair (g, x) satisfies $\sum_{e \in \gamma} x_e = \omega_\gamma - \lambda$ for each $\gamma \in \Gamma$ and $x_e(u_e - \sum_{e\gamma} g_e) = 0$ for each $e \in E$, where $\lambda \in Q$ is a given variable. These are the relaxed optimality conditions for weighted abstract flow and weighted abstract cut, where λ specifies deviated solutions from the optimum solution, [31]. The set of restricted elements is denoted by $R := \{e \in E : x_e > 0\} \subseteq E$. A solution of the restricted abstract maximum flow problem is a maximum flow g under the condition that the flow through restricted elements remains unchanged. The LP formulation of restricted maximum abstract flow is

$$\max \sum_{\gamma \in \Gamma} g(\gamma), \tag{3.10}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} g(\gamma) \leq u_e, \ \forall \, e \in E, \tag{3.11}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} g(\gamma) = u_e, \ \forall \, e \in R, \tag{3.12}$$

$$g(\gamma) \geq 0, \ \forall \, \gamma \in \Gamma. \tag{3.13}$$

The restricted minimum abstract cut problems can be written as

$$\min \sum_{e \in E} l_e u_e, \tag{3.14}$$

$$\sum_{e \in \gamma} l_e \geq \omega(\gamma), \ \forall \, \gamma \in \Gamma, \tag{3.15}$$

$$l_e \geq 0, \ \forall \, e \in E \backslash R, \text{ where } l_e \text{ is the length of } e. \tag{3.16}$$

Let g^i be an abstract dynamic flow in the abstract time expanded network with extended paths $(E^*_T, \Gamma^{*\sigma}_T)$ for a time horizon i. All paths use a counting element t^τ_c for some $\tau \in \{1, 2, \ldots, i\}$. Let $v(t^\tau_c) = \sum_{\gamma : t^\tau_c \in \gamma} g_\gamma$ be the value of flow through the counting element. Define a new abstract time-expanded network by specifying new capacities u'_e, where $u'_e = u_e$ for all elements e that are not counting elements. The capacity of counting elements is restricted to the flow value through them, i. e., we set $u'_{t^\tau_c} = v(t^\tau_c)$ for all counting elements t^τ_c in the time-expanded network. All paths use exactly one of the counting elements and they are all saturated. Thus, flow g^i remains feasible in the abstract time-expanded network with the new capacities u'_e. Define a feasible dual solution by setting

$$x_e = \begin{cases} u'_e, & e \text{ is a counting element,} \\ 0, & \text{otherwise,} \end{cases} \quad \text{for all } e \in E.$$

The flow f and dual values y are feasible in the larger abstract time expanded network $(E^*_{i+1}, \Gamma^{*\sigma}_{i+1})$ for a time horizon increased by 1. The algorithm by Martens and McCormick [154] has been applied to augment flow without removing flow on the counting elements in the process of finding restricted maximum abstract flow. In the classical setting, the Triple - Optimization - Theorem, [6] states that an earliest arrival flow is equal to a minimum cost flow where the costs equal the transit times. The same idea has been applied in abstract flow model and defined rewards such that they reflect the arrival time. A path that arrives earlier has the higher reward. Paths arriving in the first time step, i. e., paths with zero travel time, have a reward of T. The reward decreases linearly with the arrival time and paths arriving at time T have a reward of 1. Let $\gamma^{*\sigma}$ be a path in the abstract time-expanded network for time horizon T, θ be the arrival time of $\gamma^{*\sigma}$. Then the reward is defined as $w(\gamma^{*\sigma}) = T - \theta + 1$ which is supermodular.

Theorem 3.6. An approximate optimal solution for Problem 3.5 can be obtained by Algorithm 5. Moreover, the approximated arrival time is not more than two times the earliest one in the worst case.

Proof. Steps 1 and 2 are feasible by definition. Step 3 is well defined; i.e. not both paths $\overrightarrow{\gamma}$ and $\overleftarrow{\gamma}$ have to be switched at a time. This is ensured by the solution of the abstract flow in auxiliary network, [31]. Switching property cancels cycle flows [14], so that there is flow along $\overrightarrow{\gamma}$ or $\overleftarrow{\gamma}$ but not in both directions at the same time. Hence, the resulting flow from Step 3 is a feasible flow with path reversals in given evacuation network $N = (E, \Gamma, u, \theta, T, S, D)$.

Algorithm 5: Approximate Abstract Earliest Arrival Contraflow

Input : Path reversible multi-terminal abstract network $N = (E, \Gamma, u, \theta, T, S, D)$ with demands d^- and supplies d^+.

1. Construct abstract auxiliary network, $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{S}, \tilde{D})$ with new capacity and transit time functions $\tilde{u}(\tilde{\gamma}) = u(\overrightarrow{\gamma}) + u(\overleftarrow{\gamma})$ and $\tilde{\theta}(\tilde{\gamma}) = \sum_{\tilde{e} \in \tilde{\gamma}} \tilde{\theta}(\tilde{e})$.

2. Solve the problem in the auxiliary network using the greedy abstract value-approximate earliest arrival flow algorithm, [31] as follows:

   (a) Set $i = 1$ and compute an abstract flow $\tilde{g}^1$ in $(\tilde{E}_1^*, \tilde{\Gamma}_1^{*\sigma})$. Define $\lambda = 1$.

   (b) Let $(\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma})$ be the abstract time-expanded network with extended paths for time horizon $i + 1$. Define path weights $\omega(\tilde{\gamma}^{*\sigma}) = T - \tau + 1$. The updated capacities and dual values are given by

   $$u'_{\tilde{e}} = \begin{cases} v(t_c^\theta), & if\ \tilde{e} = v(t_c^\theta) \ \text{is a counting element} \\ u_{\tilde{e}}, & \text{otherwise} \end{cases} \quad \text{for all } \tilde{e} \in \tilde{E}_{i+1}^*,$$

   $$x_{\tilde{e}} = \begin{cases} \omega(\tilde{\gamma}^{*\sigma}) - \lambda, & if\ \tilde{e} \ \text{is a counting element on path } \tilde{\gamma}^{*\sigma} \in (\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma}) \\ 0, & \text{otherwise} \end{cases}$$

   for all $\tilde{e} \in \tilde{E}^{*^{i+1}}$.

   (c) Compute an abstract flow $\tilde{f}^{i+1}$ in $(\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma})$ applying the restricted abstract maximum flow minimum cut algorithm by Martens and McCormick [154].

   (d) If $\tilde{g}^{i+1}$ satisfies all balances, return flow value f^{i+1}. Else, set $i = i + 1$ and continue with 2b.

3. A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $u(\overrightarrow{\gamma})$ or there is a non-negative flow along path $\overrightarrow{\gamma} \notin \Gamma$.

Output: Earliest arrival abstract contraflow transshipment in $N = (E, \Gamma, u, \theta, T, S, D)$.

Let $\tilde{N}_T^* = (\tilde{E}_T^*, \tilde{\Gamma}_T^{*\sigma})$ be the corresponding extended abstract time expanded network of $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{S}, \tilde{D})$ with intermediate waiting, [32], where T be a significantly large time horizon. Step 2 provides the 2-value-approximate abstract earliest arrival flow in the extended abstract time expanded network of auxiliary abstract network which is an optimal flow for auxiliary network, [31]. The path capacity is increased by adding the capacity of both directions between the terminals and either direction of path is allowed with modified network. A 2-value approximate earliest arrival flow with path reversal in $N = (E, \Gamma, u, \theta, T, S, D)$ is also a feasible solution to the earliest arrival flow problem in the auxiliary network $\tilde{N} = (\tilde{E}, \tilde{\Gamma}, \tilde{u}, \tilde{\theta}, T, \tilde{S}, \tilde{D})$. As the amount of flow sent from S to D in Step 2 is not changed in Step 3, the resulting flow is the 2-value approximate solution for Problem 3.5. □

CHAPTER 4

NETWORK FACILITY LOCATION WITH CONTRAFLOW

The facility location in an evacuation network correlates the pre- and post-disaster management. It can be incorporated with contraflow approach for better evacuation planning. Our facility location contraflow solutions obtain optimal plans concerning the given and arbitrary locations to evacuate the maximum number of evacuees. The ContraFlowLoc models locate facility and maximize the flow on updated network, Section 4.3. The maximum FlowLoc over time has been introduced and solved efficiently in Section 4.2.2 (cf. Problem 4.2, Algorithm 7, Theorem 4.2, and Corollary 4.1). The maximum ContraFlowLoc integrates the contraflow with facility location problem. The solution algorithm for the problem is presented in Section 4.3.1 (cf. Problem 4.3, Algorithm 8 Theorem 4.3, and Corollary 4.2). Similarly, maximum dynamic ContraFlowLoc problem have been introduced and solved in Section 4.3.2 (cf. Problem 4.4, Algorithm 9, and Theorem 4.3). The ContraFlowloc problems have been extended and solved for the case of arbitrary locations in Section 4.4.1 and Section 4.4.2, respectively (cf. Problem 4.5, Algorithm 10, Theorem 4.5, Problem 4.6, Algorithm 11 and Theorem 4.6).

4.1 FlowLoc

The location theory was introduced in [38] with applications for industries. Different discrete network location models and algorithms have been investigated in [39]. The influence of facilities on the walking speed, the walking behaviour of pedestrians, the necessity of placing security personnels to guide the pedestrians to the locations, different positions of facilities and their influence on the behaviour, and walking speed of pedestrians have been investigated. Identifying the safe destination for the evacuation planning is an important task for the prompt transportation. Based on network flow principle, optimal sink location models and solution procedures with and without contraflow have been given in [155]. They have also illustrated their solution in Kathmandu road network. However, [41] uses location theory to improve the existing evacuation models where two different models, network flow and location theory have been integrated to introduce FlowLoc theory in evacuation modeling.

Let $\mathbb{L} \subseteq A$ be the set of all feasible locations, $\mathbb{P}$ the set of all facilities, $r : \mathbb{P} \to \mathbb{N}$ the size of the facilities and nol $: \mathbb{L} \to \mathbb{N}$ the number of facilities that can be placed on the possible locations. The FlowLoc problem asks for an allocation loc $: \mathbb{P} \to \mathbb{L}$ of the facilities to the arcs, such that the $s - t$ flow value in the network $N^{\mathbb{L}} = (V, A, u', \theta, T, s, t)$ is maximized where $u'_e = u_e - \max\{r_p : \text{loc}(p) = e\}$. If more than one facility is placed on location l only the size of the largest facility determines the reduction of the capacity on the arc. Other modelling alternatives for placing more than one facility on an edge has been discussed in [41]. The multi-facility FlowLoc problems (q-FlowLoc) find locations for the q facilities $p \in \mathbb{P}$ with size r_p such that the reduction of the maximum flow value is as small as possible and not more than nol(l) facilities are placed on each arc $l \in \mathbb{L}$. In particular, the Single-FlowLoc problem assigns one facility for $q = 1$ among the given set of facilities.

In contraflow approach, the auxiliary network of given network will be constructed by adding the capacities of two way arcs and allowing the directions in both ways with symmetric capacities and transit times. The auxiliary network $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T)$ is constructed from given evacuation network N as $\overrightarrow{e} = (v, w) \in \tilde{A}$, if $\overrightarrow{e} \in A$ or $\overleftarrow{e} = (w, v) \in A$. The arc capacity function $\tilde{u}$ is given by $\tilde{u}_{\tilde{e}} = u(\overrightarrow{e}) + u(\overleftarrow{e})$ for all arcs $\tilde{e} \in \tilde{A}$. The transit time is defined as follows

$$\tilde{\theta}(\tilde{e}) = \begin{cases} \theta(\overrightarrow{e}) & \text{for } \overrightarrow{e} \in A, \\ \theta(\overleftarrow{e}) & \text{else,} \end{cases} \quad \forall\, \tilde{e} \in \tilde{A}.$$

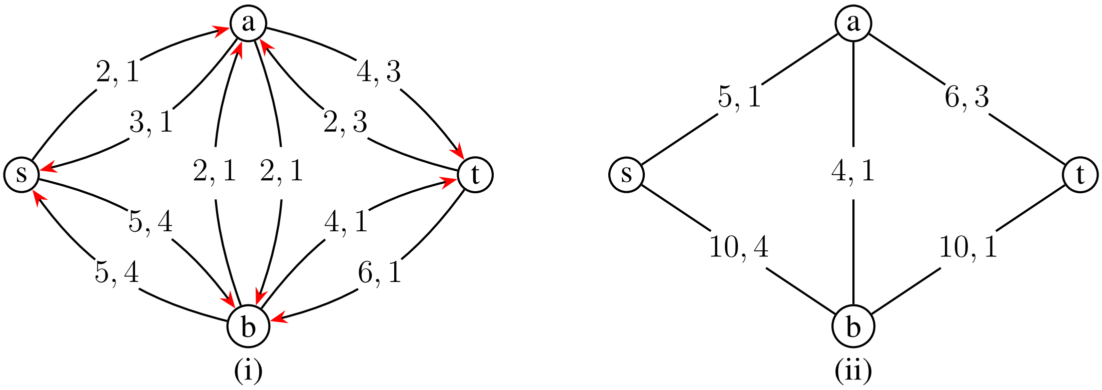


Figure 11: Evacuation and auxiliary network, respectively.

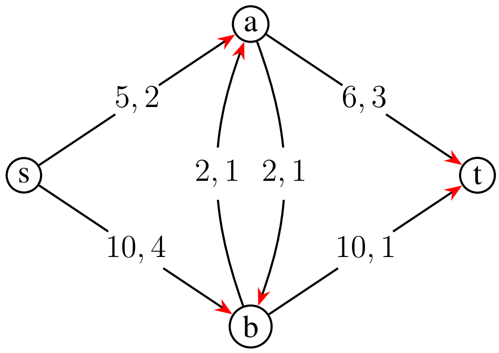


Figure 12: Reconfigured network of Figure 11(i).

4.2 FlowLoc Problems

The location of emergency units or other supports are most affecting factors in the evacuation network. Placing any facilities on arcs can affect the size of maximum flow value and quickest evacuation time. The multi-terminal q-FlowLoc problem (q-MT-FlowLoc) is $\mathcal{NP}$-complete, [156]. Authors in [156] provide some heuristic solutions for it. Here, we consider the Single-FlowLoc problem with $q = 1$ and present polynomial-time algorithms to solve it. In a recent work, the FlowLoc problem is extended to the quickest FlowLoc and solved polynomially considering single facility. A mixed integer programming formulation for multi-facility is given with its heuristics. These problems have been integrated with contraflow approach to give their ContraFlowLoc models and have been solved, [157].

4.2.1 Maximum static FlowLoc

The maximum flow can be solved by using labeling algorithm in pseudo polynomial-time $\mathcal{O}(mnU)$, where m, n and U represent number of arcs, number of vertices and maximum capacity of arcs, respectively. The shortest augmenting path algorithm finds the shortest augmenting path in polynomial-time $\mathcal{O}(m^2n)$ from the residual network considering the number of arcs as cost. Several algorithms can be found to improve the complexity of the maximum flow algorithm. We consider the maximum flow algorithm of time complexity $\mathcal{O}(nm)$, [71]. The maximum FlowLoc problem on static network and its solution procedures have been presented in [41]. They solved the problem in time $\mathcal{O}(|\mathbb{L}|n^3)$. But the same problem can be solved using Algorithm 6 in time $\mathcal{O}(|\mathbb{L}|nm)$. The maximum FlowLoc problem and its efficient solution for the two-terminal static network will be studied in this section.

Problem 4.1. The maximum FlowLoc problem on static network $N = (V, A, u_e, s, t)$ asks to locate the facility in possible locations of the network such that the resulting static flow is maximum in the updated network $N^{\mathbb{L}} = (V, A, u', s, t)$.

Algorithm 6: Maximum Static FlowLoc

Input : A directed static network $N = (V, A, u, s, t)$, locations $\mathbb{L}$, size r_p of facility p

Set: max_flow := -1
  for all $l \in \mathbb{L}$ **do**
    if $u_l \geq r_p$ **then**
      $u_l = u_l - r_p$
      max_flow_ temp = v(max_flow$(N = (V, A, u, s, t))$)
      $u_l = u_l + r_p$
      if max_flow <max_ flow_ temp **then**
        max_flow = max_flow_temp
        loc$(p) = l$
      end if
    end if
  end for
  return max_flow, loc(p)

Output: Maximum flow value max_flow, location loc(p) of facility p in the network $N = (V, A, u, s, t)$.

Theorem 4.1. The maximum FlowLoc problem can be solved optimally in time $\mathcal{O}(|\mathbb{L}|nm)$.

Proof. Algorithm 6 iterates through all possible locations $l \in \mathbb{L}$, determines the maximum flow from source to sink if location l hosts facility p and thus finds the optimal location for facility p by comparing all those maximum flow values. For every possible location, maximum flow has to be determined. Thus, Algorithm 6 has the complexity $\mathcal{O}(|L|nm)$, where $\mathcal{O}(nm)$ is the complexity of a maximum flow algorithm. Hence, Algorithm 6 solves Problem 4.1 optimally in polynomial-time. □

Example 4.1. Consider the static network given by Figure 11(i) ignoring all arc transit times. Suppose $\mathbb{L} = \{(s,a),(s,b),(a,b),(a,t),(b,t)\}$, nol $= 1$ and $r_p = 2$. Before locating facility on the network, 7 units of flow can be sent to the sink through the paths $s-a-t,\ s-b-a-t$ and $s-b-t$ with path flows $2,\ 2$, and 3, respectively. If we pick location (a,b) for the facility it will not reduce any flow from the given network. Thus, optimal flow after providing location remains the same. But it is not necessary that the optimal flow should remain the same in all networks after providing a facility on given network, for example the placement of facility on any other arcs of L reduces the flow value. Thus, the location (a, b) and flow value 7 are the optimal FlowLoc solutions.

4.2.2 Maximum dynamic FlowLoc

In this section, we introduce FlowLoc problem on dynamic network that locate given facility in given possible location and maximize the flow on updated dynamic network. We also proposed an efficient algorithm for the problem. Either temporally repeated solution, minimum cost circulation solution considering time as cost or static solution with time-expanded network can be used to find the maximum dynamic solution. The complexity of static solution with time-expanded network depends in given time horizon so that it is not polynomial. Thus, we have considered the most efficient temporally repeated solution approach which solves the maximum dynamic flow problem in time $\mathcal{O}((m \log n)(m + n \log n))$.

Problem 4.2. The maximum dynamic FlowLoc problem is to locate the facility in a possible location such that the dynamic flow is maximum in $N^{\mathbb{L}} = (V, A, u', \theta, T, s, t)$.

Algorithm 7: Maximum Dynamic FlowLoc

Input : A dynamic network $N = (V, A, u, \theta, T, s, t)$, locations $\mathbb{L}$, size r_p of the facility p

Set: max_dyna_flow := -1
for $l \in \mathbb{L}$ **do**
    if $u_l \geq r_p$ **then**
        $u_l = u_l - r_p$
        max_dyna_ flow_ temp = $v($max_dyna_flow$(N = (V, A, u, \theta, T, s, t)))$
        $u_l = u_l + r_p$
        if *max_dyna_flow* $<$ *max_dyna_flow_ temp* **then**
            max_dyna_flow = max_dyna_flow_temp
            loc$(p) = l$
        end
    end
end
return max_dyna_flow, loc(p)

Output: Maximum flow value max_dyna_flow, location loc(p) of facility p in $N = (V, A, u, \theta, T, s, t)$.

Theorem 4.2. The maximum dynamic FlowLoc problem can be solved optimally.

Proof. Algorithm 7 iterates through all possible locations $l \in \mathbb{L}$, gives the maximum dynamic flow value in updated $N^{\mathbb{L}} = (V, A, u', \theta, T, s, t)$ if location l hosts facility p in given network. The optimal location for facility p is obtained by comparing all those maximum dynamic flow values. Hence, Algorithm 7 solves Problem 4.2 optimally. □

Corollary 4.1. Algorithm 7 solves the maximum dynamic FlowLoc problem in polynomial-time complexity.

Proof. The complexity of the Algorithm 7 is $\mathcal{O}(|L|(m \log n)(m + n \log n))$, where $\mathcal{O}((m \log n)(m + n \log n))$ represents the complexity of maximum dynamic flow. □

Example 4.2. Consider Figure 11(i) with $\mathbb{L} = \{(s,a),(s,b),(a,b),(a,t),(b,t)\}$, nol $=$ 1, $r_p = 2$ and $T = 8$. The solution can be obtained by applying Algorithm 6. The optimal flow is 27 with optimal location (a,b), shown in Table 1.

Table 1: Maximum dynamic flow corresponding to location (a,b).

Time horizon	Paths	Length of path	Flow	Total dynamic flow
4	$s-a-t$	4	2	2
5	$s-a-t$	4	2	
	$s-b-t$	5	4	8
6	$s-a-t$	4	2	
	$s-b-t$	5	4	14
7	$s-a-t$	4	2	
	$s-b-t$	5	4	20
8	$s-a-t$	4	2	
	$s-b-t$	5	4	
	$s-b-a-t$	8	1	27

The maximum dynamic flow problem maximizes the solution at given time horizon, where as the earliest arrival flow problem maximizes the solution at every possible time from the beginning. This implies that, every earliest arrival solution is maximum dynamic solution but the converse may not be true. The earliest arrival flow problem has been solved by using the successive shortest augmenting path algorithm in [121]. The standard chain decomposition could not be adopted to solve the earliest arrival problem as we can not repeat the same path for each time. The non-standard chain decomposition uses the minimum cost maximum flow approach assuming backward chain flows. We can not fixed the direction of any arc to get the earliest arrival flow. Thus, the FlowLoc problem could be extended to different dynamic flow problems in various networks, but it can not be extended to the earliest arrival flow models. For the justification we provide following example.

Example 4.3. Consider Figure 12 with $\mathbb{L} = \{(s,a),(s,b),(a,b),(a,t),(b,t)\}$, nol $= 1$, and $r_p = 2$. The earliest arrival flow without any facilities on the network is 65, as shown in Table 2. Applying successive shortest path algorithm on the network, different solutions with respect to different locations have been shown in Table 3 and Table 4. From the table, facility on the location (a, b) reduce the minimum flow but it is not earliest arrival flow, since 2 units of flow can be send in $T = 3$. If we look for earliest arrival flow we loss the maximality in some interval of time. Again, facility on locations $\{(s, a), (s, b), (a, t), (b, t)\}$ will send 2 units of flow in $T = 3$ but could not send as much flow as we send by considering the location (a, b) from some point of time. For example, if we consider location (a, t) we can send 33 units of flow in $T = 6$ but by considering location (a, b), 35 units of flow can be sent in the same time.

Table 2: Earliest arrival flow on network given in Figure 12.

Time horizon	Paths	Length of path	Flow	Earliest arrival flow
3	$s-a-b-t$	3	2	2
4	$s-a-b-t$	3	2	
	$s-a-t$	4	3	7
5	$s-a-t$	4	3	
	$s-b-t$	5	10	20
6	$s-a-t$	4	5	
	$s-b-t$	5	10	35
7	$s-a-t$	4	5	
	$s-b-t$	5	10	50
8	$s-a-t$	4	5	
	$s-b-t$	5	10	65

4.3 ContraFlowLoc Problems

In this section, we define both static and dynamic maximum ContraFlowLoc problems and present efficient algorithms to solve them. The proposed maximum ContraFlowLoc algorithms solve the problems in the same time complexity as the maximum FlowLoc problems but the value of flow can be double after contraflow reconfiguration.

Table 3: Earliest arrival FlowLoc on the network shown in Figure 12.

Locations	Time horizon	Paths	Length of path	Flow	Total EAF
(s, a)	3	$s-a-b-t$	3	2	2
	4	$s-a-b-t$	3	2	
		$s-a-t$	4	1	5
	5	$s-a-t$	4	1	
		$s-b-t$	5	10	16
	6	$s-a-t$	4	3	
		$s-b-t$	5	10	29
	7	$s-a-t$	4	3	
		$s-b-t$	5	10	42
	8	$s-a-t$	4	3	
		$s-b-t$	5	10	55
(s, b)	3	$s-a-b-t$	3	2	2
	4	$s-a-b-t$	3	2	
		$s-a-t$	4	3	7
	5	$s-a-b-t$	3	2	
		$s-a-t$	4	3	
		$s-b-t$	5	8	20
	6	$s-a-b-t$	3	2	
		$s-a-t$	4	3	
		$s-b-t$	5	8	33
	7	$s-a-b-t$	3	2	
		$s-a-t$	4	3	
		$s-b-t$	5	8	46
	8	$s-a-b-t$	3	2	
		$s-a-t$	4	3	
		$s-b-t$	5	8	59

Table 4: Remaining part of Table 3.

Locations	Time horizon	Paths	Length of path	Flow	Total EAF
(a,t)	3	$s-a-b-t$	3	2	2
	4	$s-a-b-t$	3	2	
		$s-a-t$	4	3	7
	5	$s-a-t$	4	3	
		$s-b-t$	5	10	20
	6	$s-a-t$	4	3	
		$s-b-t$	5	10	33
	7	$s-a-t$	4	4	
		$s-b-t$	5	10	47
	8	$s-a-t$	4	4	
		$s-b-t$	5	10	61
(a,b)	4	$s-a-t$	4	5	5
	5	$s-a-t$	4	5	
		$s-b-t$	5	10	20
	6	$s-a-t$	4	5	
		$s-b-t$	5	10	35
	7	$s-a-t$	4	5	
		$s-b-t$	5	10	50
	8	$s-a-t$	4	5	
		$s-b-t$	5	10	65

4.3.1 Maximum static ContraFlowLoc

The maximum static FlowLoc problem locates the facility on possible locations and maximize the flow on updated network whereas the contraflow allows arc reversals to improve the solution. The maximum FlowLoc problem is introduced in [41] and the maximum contraflow problem in [16]. We define the maximum ContraFlowLoc problem in static network. We also propose an efficient algorithm to solve the Problem 4.3.

Table 5: Remaining part of Table 4.

Locations	Time horizon	Paths	Length of path	Flow	Total EAF
(b,t)	3	$s-a-b-t$	3	2	2
	4	$s-a-b-t$	3	2	
		$s-a-t$	4	3	7
	5	$s-a-t$	4	3	
		$s-b-t$	5	8	18
	6	$s-a-t$	4	5	
		$s-b-t$	5	8	31
	7	$s-a-t$	4	5	
		$s-b-t$	5	8	44
	8	$s-a-t$	4	3	
		$s-b-t$	5	10	
		$s-b-a-t$	8	2	59

Problem 4.3. Given network $N = (V, A, u, s, t)$, locations $\mathbb{L}$ and size r_p of facility p, the maximum static ContraFlowLoc problem finds the maximum flow in $N^{\mathbb{L}} = (V, A, u', s, t)$ providing efficient location for the facility with arc reversals capability.

Algorithm 8: Maximum Static ContraFlowLoc

Input : A static network $N = (V, A, u, s, t)$, locations $\mathbb{L}$, size r_p of facility p.

1 Construct auxiliary $\tilde{N} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$.

2 Apply Algorithm 6 in $\tilde{N} = (V, \tilde{A}, \tilde{u}, s, t)$ considering locations $\mathbb{L}$.

3 Decompose the maximum flow resulting from Step 2 into chain and cycle flows then remove the cycle flow.

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u'(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$.

Output: Maximum contraflow value max_cont, location loc(p) of facility p in $N = (V, A, u, s, t)$.

Theorem 4.3. Algorithm 8 solves the maximum ContraFlowLoc problem optimally.

Proof. The Steps 1 and 2 are feasible by definition. If $r_p > u_{\overrightarrow{e}}$ and $loc(p) = \overrightarrow{e}$ on the auxiliary network then the capacity of $\overleftarrow{e}$ is defined by $u'(\overleftarrow{e}) = u(\overleftarrow{e}) + u(\overrightarrow{e}) - r_p$, since the facility will occupy all capacity of $\overrightarrow{e}$ and remaining from the capacity of $\overleftarrow{e}$. Thus, Step 4 is well defined, i.e. not both arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be reversed at a time which is ensured by using Step 3. There is flow along $\overrightarrow{e}$ or $\overleftarrow{e}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with arc reversals in $N = (V, A, u, s, t)$.

Algorithm 6 has been used in Step 3 for the solution on $\tilde{N} = (V, \tilde{A}, \tilde{u}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$. In fact, any optimal solution to the FlowLoc maximum flow problem with arc reversals on $N = (V, A, u, s, t)$ is also a feasible solution to the maximum FlowLoc problem on $\tilde{N} = (V, \tilde{A}, \tilde{u}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$. Moreover, the amount of flow sent from s to t in Steps 2 is not changed in Step 4, the resulting solution is an optimal for the Problem 4.3. □

Corollary 4.2. Algorithm 8 finds optimal solution for the maximum ContraFlowLoc problem in polynomial-time.

Proof. The direction of paths can be reversed using Step 3 in linear time thus the construction of auxiliary network takes linear time. Thus, the complexity of Algorithm 8 depends on the complexity of Step 2. Hence, Algorithm 8 finds optimal solution for the Problem 4.3 in polynomial-time, Theorem 4.3. □

Example 4.4. As in Example 4.1, let $\mathbb{L} = \{(s, a), (s, b), (a, b), (a, t), (b, t)\}$ be the locations, nol $= 1$ and $r_p = 2$. Using contraflow approach on the same network, 15 units of flow can be send to the sink through the paths $s-a-t$ and $s-b-t$ with path flows 5 and 10, respectively. Thus, the optimal flow value is 15 with optimal location (a, b). Note that the optimal locations could be changed after using contraflow technique. However, the optimal location remains the same in this example.

4.3.2 Maximum dynamic ContraFlowLoc

The maximum dynamic ContraFlowLoc problem is introduced to locate the facility on given network such that the loss in maximum dynamic contraflow is minimum on given network, i.e., the dynamic flow on updated network is maximum. We also present an efficient algorithm for solving Problem 4.4.

Problem 4.4. Given a network $N = (V, A, u, \theta, T, s, t)$, locations $\mathbb{L}$ and size r_p of facility p, the maximum dynamic ContraFlowLoc problem in $N = (V, A, u, \theta, T, s, t)$ is to find the maximum dynamic flow on updated network $N^{\mathbb{L}} = (V, A, u', \theta, T, s, t)$ with arc reversals allowed initially.

Algorithm 9: Maximum Dynamic ContraFlowLoc

Input : A dynamic network $N = (V, A, u, \theta, T, s, t)$, locations $\mathbb{L}$, size r_p of facility p.

1 Construct auxiliary $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and symmetric transit time.

2 Apply Algorithm 7 in $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ considering locations $\mathbb{L}$.

3 Decompose the maximum dynamic flow resulting from Step 3 into chain and cycle flows then remove the cycle flow.

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u'(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$.

Output: Maximum dynamic contraflow value max_dyna_cont, location $\text{loc}(p)$ of facility p in $N = (V, A, u, \theta, T, s, t)$.

Theorem 4.4. The maximum dynamic ContraFlowLoc problem can be solved optimally in time $O(|L|(m \log n)(m + n \log n))$.

Proof. As in Theorem 4.3, Steps 1, 2 and 4 are feasible and well defined. The solution on $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ has been obtained using Algorithm 7 in Step 2. Indeed, any optimal solution to the problem with arc reversals on $N = (V, A, u, \theta, T, s, t)$ is also a feasible solution to the problem on $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$. Every feasible flow of the maximum dynamic FlowLoc problem in $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ is feasible to the maximum dynamic ContraFlowLoc problem in $N = (V, A, u, \theta, T, s, t)$. The maximum dynamic contraflow with facility in $N = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ is not more than the maximum dynamic flow in $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ with facility. The maximum dynamic flow can be obtained by temporally repeating a path flow of a static graph, [1]. Since the amount of flow sent from s to t in Steps 3 is not changed in Step 4, the resulting FlowLoc solution is optimal for the Problem 4.4. As described in Corollary 4.2, the running time of the Algorithm 9 depends on the running time of Step 3, indeed in the running time of the Algorithm 7. Hence, the Algorithm 9 solves the Problem 4.4 in polynomial time, Corollary 4.1. □

Example 4.5. Apply the Algorithm 9 on the network of Figure 11(i) for $T = 8$, $r_p = 2$ and $\mathbb{L} = \{(s,a),(s,b),(a,b),(a,t),(b,t)\}$. The optimal flow is 65 which can be send through the paths $s-a-t$ and $s-b-t$ in different time intervals providing optimal location (a,b) (see Table 2 corresponding to location (a,b)). Here, the optimal flow value after contraflow is more than the double flow value before contraflow configuration.

The earliest arrival flow model has been integrated with contraflow to introduce earliest arrival contraflow in [18]. They have presented different algorithms to get exact and approximate solutions. According to the idea of FlowLoc presented in Section 4.2.2 and Example 4.3 we are not combining the earliest arrival contraflow model with location analysis to define the FlowLoc earliest arrival contraflow problem.

4.4 ContraFlowLoc Problems for Extended Locations

The FlowLoc problems with given possible facility locations are studied in Section 4.3. In this section we consider a FlowLoc problem, where a location will be feasible if its size is at least the respective arc capacity. The problem is modeled with and without the lane reversal permissibility. In both cases, efficient algorithms are presented. In this approach, the contraflow configuration increases the number of facilities with modified arc capacities.

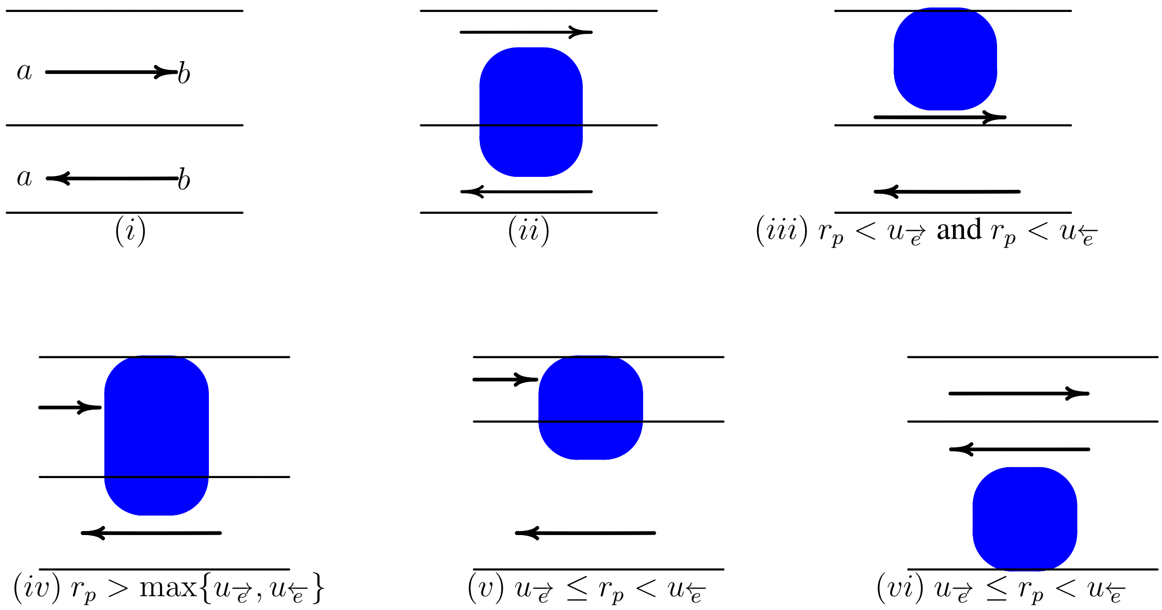


Figure 13: Variations of facility locations on auxiliary network for the modified capacities of arcs and facility locations determined by its size.

Consider Figure 13(i), which represents two way arcs (a, b) and (b, a). The maximum ContraFlowLoc solution locates a facility on the auxiliary network and finds an optimal flow in the network with reduced capacity by r_p. We exclude of locating a facility on the middle of a road as shown in Figure 13(iii). We consider the following cases.

1. Suppose the facility is located on (a, b) as shown in Figure 13(ii). If a maximum contraflow is in the direction of (a, b) and it is greater than $u'(a, b)$, then the direction of arc (b, a) has to be reversed.

2. Suppose the capacity of facility is greater than the capacities of both arcs (a, b) and (b, a), then the arc reversal problem reduces to the single arc problem. In this situation, the facility can be located on any side, for example as shown in Figure 13(iv).

3. Suppose the capacity of only one arc is sufficient to allocate facility but not its backward arc, then the problem becomes as shown in Figures 13(v) and 13(vi). The former is as in Figure 13(iv) and the latter is as in Figure 13(ii).

Example 4.6. Let $r_p = 3$, $u(a,b) = 2$, and $u(b,a) = 4$ on a single arc of a network. Suppose that we locate facility on (a,b) in the auxiliary network and a maximum-flow solution in the auxiliary network yields a flow 2 along the direction of (a,b). Then, the arc (b,a) should be reversed. But if we would have located the facility on (b,a) in the auxiliary network then there was no need to reverse the above arc to pass this flow value. It matches to the case of Figure 13 (v-vi). Indicating that a wrong decision may lead to unnecessary arc reversals.

4.4.1 Maximum static ContraFlowLoc

The maximum static ContraFlowLoc problem and its solution procedure has been presented in this section.

Problem 4.5. Given network $N = (V, A, u, s, t)$, locations $\mathbb{L} = \{e \in A \mid u_e \geq r_p\}$, where r_p denotes the size of facility p, the maximum static ContraFlowLoc problem is to find the maximum flow in $N^{\mathbb{L}} = (V, A, u', s, t)$ with arc reversal capability.

Algorithm 10: Maximum Static ContraFlowLoc with Extended Locations

---

Input : A static network $N = (V, A, u, s, t)$, locations $\mathbb{L} = \{e \in A \mid u_e \geq r_p\}$ where r_p denotes the size of facility p.

1 Construct auxiliary network $\tilde{N} = (V, \tilde{A}, \tilde{u}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and locations $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$.

2 Apply Algorithm 6 in $\tilde{N} = (V, \tilde{A}, \tilde{u}, s, t)$ considering modified set of locations $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$.

3 Decompose the maximum flow resulting from Step 2 into chain and cycle flows then remove the cycle flow.

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$.

5 If facility is located on $\tilde{e} \in \tilde{A}$ and the flow is positive along the direction of the arc $\overrightarrow{e} \in A$ then allocate the facility along the direction of $\overleftarrow{e} \in A$.

Output: Maximum contraflow value max_cont, location $\text{loc}(p)$ of facility p in $N = (V, A, u, s, t)$.

Theorem 4.5. The maximum ContraFlowLoc problem with extended locations can be solved optimally in time $\mathcal{O}(|\tilde{\mathbb{L}}|nm)$.

Proof. The construction of possible location set $\mathbb{L} = \{e \in A \mid u_e \geq r_p\}$ is feasible as the facility will host in any location if it has sufficient capacity. Step 1 constructs the auxiliary network with new location set $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$ and capacity $\tilde{u}$ that are feasible. Step 2 is feasible as it calculates FlowLoc solution in the auxiliary network considering location $\tilde{\mathbb{L}}$. Step 4 is well defined, i.e. not both arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be reversed at a time, this is ensured by the solution in auxiliary network. The resulting flow is decomposed into path and cycle flows and the cycle flows are removed in Step 3. So that there is flow along $\overrightarrow{e}$ or $\overleftarrow{e}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with arc reversals in $N = (V, A, u, s, t)$.

Optimal locations for the facility will be obtained in Step 2 for auxiliary network. If there are two way arcs on given network and the auxiliary arcs host the facility on auxiliary network then Step 5 decides the location of the facility in the given network according to the direction of flow provided from Step 3. Thus, Step 5 is feasible. Further proof of this theorem can be completed as in Theorem 4.3. □

Example 4.7. Let nol $= 1$ and $r_p = 4$ with ignored travel times in Figure 11(i). Then, $\mathbb{L} = \{(a, t), (s, b), (b, s), (b, t), (t, b)\}$ be the possible locations. Applying Algorithm 6, we can send 7 units of flow before contraflow. In this case, both locations (b, s) or (t, b) are the optimal location as the resulting flow remains the same. The arc (b, a) or (a, b) was not able to host the facility due to low capacities.

But using contraflow approach, both arcs have been combined to increase the capacity. As a result, all arcs become feasible to host the facility with capacity 4. After locating facility on the arc, 15 units of flow can be send to the sink through the paths $s-a-t$ and $s-b-t$ with path flows 5 and 10, respectively. Thus, the optimal flow value is 15 with optimal location (a, b). Here, the facility on the network did not affect the optimality.

4.4.2 Maximum dynamic ContraFlowLoc

The maximum dynamic ContraFlowLoc problem with capacity constrained locations and its procedure has been introduced in this section.

Problem 4.6. Given a network $N = (V, A, u, \theta, T, s, t)$, locations $\mathbb{L} = \{e \in A \mid u_e \geq r_p\}$, where r_p denotes the size of facility p, the maximum dynamic ContraFlowLoc problem on $N = (V, A, u, \theta, T, s, t)$ is to find the maximum dynamic flow on updated network $N^{\mathbb{L}} = (V, A, u', \theta, T, s, t)$ with arc reversals allowed initially.

Algorithm 11: Maximum Dynamic ContraFlowLoc with Extended Locations

---

Input : A dynamic network $N = (V, A, u, \theta, T, s, t)$, locations $\mathbb{L} = \{e \in A \mid u_e \geq r_p\}$, size r_p of facility p.

1 Construct auxiliary $\tilde{N}^{\mathbb{L}} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ with new locations $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$, capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and symmetric transit time $\tilde{\theta}$.

2 Apply Algorithm 7 in $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ for locations $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$.

3 Construct the temporally repeated flow resulting from Step 2.

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$.

5 If facility is located on $\tilde{e} \in \tilde{A}$ and the flow is positive along the direction of the arc $\overrightarrow{e} \in A$ then allocate the facility along the direction of $\overleftarrow{e} \in A$.

Output: Maximum dynamic contraflow value max_dyna_cont, location loc(p) of facility p in $N = (V, A, u, \theta, T, s, t)$.

Theorem 4.6. Algorithm 11 solves Problem 4.6 optimally in $\mathcal{O}(|\tilde{\mathbb{L}}|(m \log n)(m+n \log n))$.

Proof. The feasibility of Algorithm 11 can be proved as in Theorem 4.5. The solutions on $\tilde{N} = (V, \tilde{A}, \tilde{u}, \tilde{\theta}, T, s, t)$ with locations $\tilde{\mathbb{L}} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$ have been calculated using Algorithm 7 in Step 2. The maximum dynamic flow obtained from Step 2 has been decomposed into temporally repeated path flows in Step 3 which is optimal flow for the Problem 4.6. The remaining part of the proof is the direct consequence of Theorem 4.4. □

Example 4.8. Consider the network of Figure 14(i) with values from Table 6 and $T = 9$. Without providing facility on network, maximum 61 and 137 units of flows can be send before and after contraflow, respectively. Our objective is to locate a facility on arc with minimum flow loss in the given network. Suppose, nol $= 1, r_p = 10$, and $T = 9$. Before contraflow, possible locations are $\mathbb{L} = \{(s,a),(s,d),(c,t)\}$. Construct the auxiliary network (shown in Figure 14 (ii)) with values of Table 7. Then, possible locations set becomes $\tilde{\mathbb{L}} = \{(s,a),(s,d),(c,t),(d,a),(d,e),(b,t)\}$. Algorithm 11 locates facility on given evacuation network of Figure 14 (i) with minimum flow loss. Different maximum dynamic flow solutions after providing facility on different locations with different time horizons are shown in Table 8. Before contraflow, the arc (d,a) or (a,d) was not feasible to host the facility due to low capacities. But using contraflow approach, arc capacity has been increased by reversing the direction. As a result, the arc become feasible to host the facility and it is optimal location for the problem. If we consider different time horizon less than the transit time of longest path (longest in the sense of transit time) optimal location can be different for different time horizons, details is shown in Table 8.

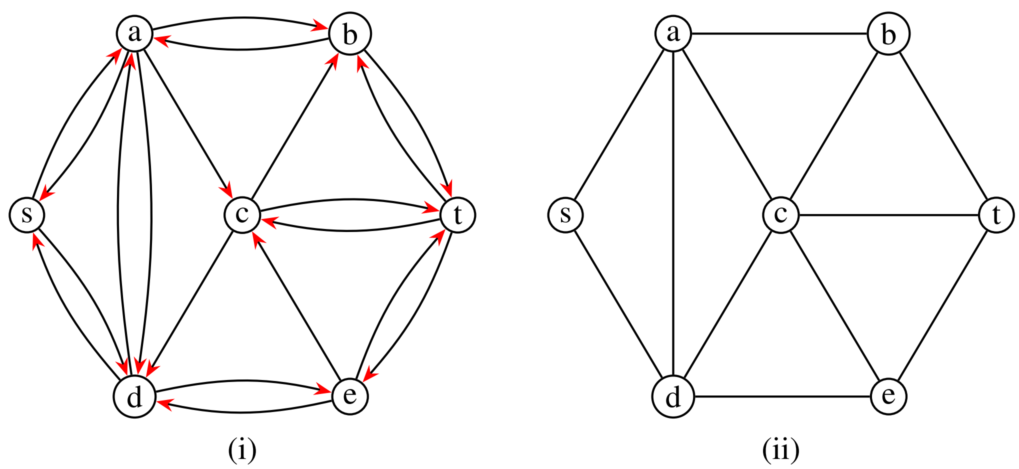


Figure 14: Dynamic network and its auxiliary network.

Based on the above concepts, the FlowLow and ContraFlowLoc problem can also be extended to other network settings. Moreover, the ContraFlowLoc models can be extended to partial ContraFlowLoc and solved with the same complexity as in ContraFlowLoc. The partial ContraFlowLow model only reverses the required arc capacities and saves the unused arcs. The saved arc can be used for the emergencies.

Table 6: Capacities and transit times of arcs corresponding to Figure 14 (i).

Arcs	(s,a)	(a, s)	(s, d)	(d, s)	(d, a)	(a,d)	(a, c)	(c, d)	(a, b)	(b, a)
Capacities	10	5	10	7	7	4	3	5	5	3
Transit times	1	1	2	2	1	1	2	2	1	1
Arcs	(c, b)	(e, c)	(d, e)	(e, d)	(c, t)	(t, c)	(b, t)	(t, b)	(e, t)	(t, e)
Capacities	6	5	8	3	10	5	2	9	4	4
Transit times	1	1	2	2	1	1	1	1	3	3

Table 7: Capacities and transit times of arcs corresponding to Figure 14(ii).

Arcs	(s,a)	(s, d)	(d, a)	(a, c)	(c, b)	(d, c)	(a, b)	(e, c)	(d, e)	(t, c)	(b, t)	(e, t)
Capacities	15	17	11	3	6	5	8	5	11	15	11	8
Times	1	2	1	2	1	2	1	1	2	1	1	3

Table 8: Variants of FlowLoc solutions for Network 14 considering the parameters of Table 6 and 7.

T	Before contraflow				After contraflow							Diff. of two optimal flows
	MDF after placing facility on			Location decisions	Maximum dynamic flow(MDF) after placing facility on						Location decisions	
	(s, a)	(s, d)	(c, t)		(s, a)	(s, d)	(c, t)	(d, a)	(d, e)	(b, t)		
3	0	2	2	(s, d)/ (c, t)	5	8	8	8	8	1	(s, d)/(c, t)/ (d, a)/(d, e)	6
4	0	7	4	(s, d)	10	19	19	19	19	5	(s, d)/(c, t) (d, a)/(d, e)	11
5	2	12	6	(s, d)	23	35	32	31	35	14	(s, d)/ (d, e)	23
6	12	21	8	(s, d)	44	56	48	52	53	28	(s, d)	35
7	22	30	14	(s, d)	66	78	72	79	70	48	(d, a)	49
8	32	39	20	(s, d)	88	100	96	106	87	68	(d, a)	67
9	42	48	26	(s, d)	110	122	120	133	104	88	(d, a)	85

CHAPTER 5

FLOW IMPROVEMENT WITH SWITCHING COSTS

An evacuation planning problem aims to shift the maximum number of evacuees from a danger areas to a safe areas in limited time. The improvement of network topology plays an important role to improve the optimal solution. But, the improvement could be affected by available incremental budget. Here, different flow improvement strategies for fixed switching costs will be investigated. Different maximum static flow improvement problems have been studied and their complexities are given in Section 5.1. Section 5.1.2 presents and solves the mathematical model for the maximum dynamic flow improvement problem (cf. Formulation 5.10–5.17, Problem 5.2, and Theorem 5.5). The contraflow reconfiguration problems with arc switching cost have been introduced in Section 5.2. Based on this approach, the maximum static contraflow problem has been introduced and solved in Section 5.2.1(cf. Problem 5.3, Algorithm 12, and Theorem 5.6). The maximum dynamic contraflow problem has been introduced and solved in Section 5.2.2. (cf. Problem 5.4, Algorithm 13, and Theorem 5.10).

5.1 Maximum Flow Improvement Problems

Different types of network modification problems exist in literature. Generally, the original network is assumed to be not modifiable in the sense that capabilities or costs remain fixed as in the given network. However, this assumption is not valid in many real evacuation scenarios. For example, the capacity of an arc can also be increased up to some limit subject to some capacity incremental cost. For this, a fixed budget can be distributed to increase capacities in the network such that the network topology is modified and an objective, for instance, the flow, with respect to new capacities are maximized. There are three variants of this improvement strategy that deal with rational, integral, and either of the all possible or not at all capacity values that are studied in [42]. The first two variants are polynomial-time solvable, while the last one is $\mathcal{NP}$-hard even in the cases of bipartite and series parallel graphs. This third variant problem, called the $0/1$ maximum flow improvement strategy, is equivalent to the maximum flow fixed cost problem which is a bicriteria optimization problem where the flow has to be maximized under the budget constraints. A fully polynomial-time approximation scheme for series parallel graphs is presented. The network modification problems that relate to arc based improvement and node based upgrading models are also investigated, as mentioned in [42].

5.1.1 Static flow improvements

In maximum static flow improvement problem (MSFIP), additional non-negative number U_e, for each $e \in A$ are given so that the capacity u_e can be increased with some non-negative cost up to the upper bound $U_e \geq u_e$. The improvement capacity function with non-negative unit cost b_e is defined as $I : A \longrightarrow \mathbb{Q}_{\geq 0}$. The objective of the problem is to maximize the flow from the sources to sinks by increasing the capacities of arcs within the budget restriction where incremental cost is to be accepted to increase arc capacity. The flow improvement problem (5.1 - 5.7) has been formulated as in [42].

$$\sum_{e\in A_v^+} f_e - \sum_{e\in A_v^-} f_e \;=\; \begin{cases} h_v, & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \\ -h_v, & \text{if } v \in V^-, \end{cases} \tag{5.1}$$

$$h_v \;\geq\; 0,\ \forall\ v \in V, \tag{5.2}$$

$$u_e + I_e \geq f_e \;\geq\; 0,\ \forall\ e \in A, \tag{5.3}$$

$$U_e \geq u_e + I_e \;\geq\; 0,\ \forall\ e \in A, \tag{5.4}$$

$$I_e \;\geq\; 0,\ \forall\ e \in A, \tag{5.5}$$

$$\sum_{e\in A} I_e b_e \;\leq\; B. \tag{5.6}$$

Constraint (5.3) controls the arc flows, Constraint (5.4) limits capacity increment and Constraint (5.6) bounds the total capacity incremental cost B with b as the unit cost. By denoting F_{st} to be the sum of flow out from the sources that enter into the sinks, the objective function equals to

$$F_{st} = \sum_{v\in V^+} h_v. \tag{5.7}$$

Problem 5.1. The MSFIP with capacity improvement cost is the maximum static flow problem (5.1-5.7), where capacity of arcs with unit costs can be increased up to specified limit bounded by the improvement cost.

The MSFIP with continuous improvement strategy in which the improvement function I_e takes any rational values respecting the upper bound, can be solved optimally in polynomial-time. The integral MSFIP that takes only integral improvements can be transformed into a budget constraint minimum cost flow problem in polynomial-time.

Assume that the unit cost b_e of increasing the capacity of arc e is non-negative number and the optimal flow improvement corresponding to the arc e is $I_e^* = \max\{0, f_e^* - u_e\}$, where f^* is the optimal improved maximum flow. Otherwise, the strategy would waste the cost. Such a behavior can be modeled by a flow cost c_e defined as,

$$c_e(f) := \begin{cases} 0 & \text{for } 0 \leq f \leq u_e, \\ b_e.(f - u_e) & \text{for } u_e < f \leq U_e, \end{cases} \quad \forall\ e \in A. \tag{5.8}$$

By definition, c_e is a piecewise-linear convex function. In this case, each $e \in A$ is replaced by two parallel arcs e_0 and e_1 to make it linear. The capacities $\bar{u}_e$ and linear costs $\bar{c}_e$ of these arcs are set as,

$$\begin{aligned} \overline{u}_{e_0} &:= \overline{u}_e, & \overline{u}_{e_1} &:= U_e - u_e, \\ \overline{c}_{e_0} &:= 0, & \overline{c}_{e_1} &:= b_e. \end{aligned} \tag{5.9}$$

This construction is valid from the convexity of cost function. Thus, the improved maximum flow value F^*_{st} can be obtained solving minimum cost flow with cost at most B by the method of binary search in $[0, nU_{max}]$, where $U_{\max} := \max\{U_e : e \in A\}$ is the maximum capacity. As a result, they established,

Theorem 5.1. [42] The integral MSFIP can be solved optimally in polynomial time by $\mathcal{O}(\log(nU_{\max}))$ minimum cost flow computations in a directed network with $2m$ arcs.

Instead of performing a binary search on the interval $[0, nU_{max}]$, it can be searched the interval only in multiplicative steps of $1 + \epsilon$, where $\epsilon > 0$ is a fixed accuracy parameter. The value F'_{st} found by this modified binary search satisfies $F'_{st} \geq \frac{F^*_{st}}{1+\epsilon}$.

Theorem 5.2. [42] Consider $\epsilon > 0$ be fixed. Then an $(1 + \epsilon)$-approximate solution for MSFIP can be obtained in $\mathcal{O}(\log \log_{1+\epsilon}(nU_{\max}))$ minimum cost flow computations in the directed graph with $2m$ arcs.

Each of the minimum cost flow computations have to be carried out in a graph with $\mathcal{O}(\xi m)$ arcs, where ξ is the maximum number of breakpoints occurring in the piecewise-linear cost functions. Furthermore, one can solve the problem in strongly polynomial-time applying Megiddo's parametric search, [48]. The same search can be applied to solve the rational MSFIP, too.

The flow improvement strategy, either to increase the capacity of each arc to its maximum capacity or leave the capacity of arc unchanged is $\mathcal{NP}$-hard. This $0/1$ MSFIP is equivalent to the maximum flow problem with fixed cost on arcs. For given non-negative capacity u_e and cost c_e on each $e \in A$, the latter problem asks to find a subset A^* of A such that $\sum_{e \in A^*} c_e \leq B$ and the source-sink flow is maximized. But, the decision variant of this problem is $\mathcal{NP}$-complete. We state, *Theorem* 5.3. [42] The maximum flow problem with fixed cost on arcs is $\mathcal{NP}$-hard even on series-parallel and bipartite graphs.

The pseudo polynomial-time algorithm for the maximum fixed cost flow problem on series-parallel graphs are presented and they are converted into a fully polynomial-time approximation scheme by scaling technique.

Theorem 5.4. [42] When the problems are restricted to series parallel graphs, the maximum flow problem with fixed cost on arcs and $0/1$ MSFIP can be solved with fully polynomial time approximation scheme.

5.1.2 Dynamic flow improvement

This section extends the MSFIP to the maximum dynamic flow improvement problem (MDFIP) considering the time factor. Let $I(\tau)$ and $b(\tau)$ be capacity improvement and improvement unit cost functions in dynamic network, respectively. Then the proposed MDFIP can be formulated as follows.

$$\max \sum_{\tau=0}^{T} \sum_{v \in V^+} h_v(\tau), \tag{5.10}$$

$$\text{s. t.} \sum_{e \in A_v^+} f_e(\tau) - \sum_{\substack{e \in A_v^- \\ \tau-\theta_e \geq 0}} f_e(\tau - \theta_e) = \begin{cases} h_v(\tau), & \text{if } v \in V^+, \\ 0 & \text{if } v \in V^0, \ \forall\, \tau \in \mathcal{T}, \\ -h_v(\tau) & \text{if } v \in V^-, \end{cases} \tag{5.11}$$

$$h_v(\tau) \geq 0, \ \forall\, v \in V, \ \forall\, \tau \in \mathcal{T}, \tag{5.12}$$

$$f_e(\tau) = 0, \ \forall\, e \in A, \ \tau = \overline{T - \theta_e + 1, \ T}, \tag{5.13}$$

$$u_e(\tau) + I_e(\tau) \geq f_e(\tau) \geq 0, \ \ \forall\, \tau \in \mathcal{T}, \forall\, e \in A, \tag{5.14}$$

$$U_e(\tau) \geq u_e(\tau) + I_e(\tau) \geq 0, \ \ \forall\, \tau \in \mathcal{T}, \forall\, e \in A, \tag{5.15}$$

$$I_e(\tau) \geq 0, \ \ \forall\, e \in A, \ \forall\, \tau \in \mathcal{T}, \tag{5.16}$$

$$\sum_{\tau=0}^{T} \sum_{e \in A} I_e(\tau) b_e(\tau) \leq B. \tag{5.17}$$

Constraints (5.14) deal with the feasibility of flow with the improved capacity and Constraints (5.15) deal with the maximum possible improvement. The budget Constraint (5.17) has been imposed to bound the capacity improvement cost.

Let $(v(\tau), w(\tau + \theta_e)) \in A_T$ and let $I^{e(\tau)}$ be the capacity improvement on arc $e \in A$, then the capacity improvement function could be defined as $I_T^{e(\tau)} := I_{(v(\tau),w(\tau+\theta_e))} = I_e(\tau),$ for all $e(\tau) \in A_T$. Similarly, the capacity improvement cost function could be defined as $b_T^{e(\tau)} := b_{(v(\tau),w(\tau+\theta_e))} = b_e(\tau),$ for all $e(\tau) \in A_T$. Thus, each dynamic network can be transformed into the corresponding time-expanded network for a given time horizon. The maximum dynamic flow in the given network is equal to the maximum flow in time-expanded network, [11].

Problem 5.2. The MDFIP with capacity improvement cost is the MDF, where capacities of arcs can be increased up to specified limit accepting the improvement cost.

Theorem 5.5. The integral and continuous flow improvement problems in dynamic network can be solved optimally in pseudo polynomial-time.

Proof. Let us transform the given integral MDFIP in dynamic network into the MSFIP in the corresponding static network assuming $V_T := \{v(\tau) : v \in V, \ \tau \in T\}$; $A_T := \{(v(\tau), w(\tau + \theta_e)) : e = (v, w) \in A, 0 \leq \tau \leq T - \theta_e\}$, $u_T^{e(\tau)} := u_e(\tau)$, $I_T^{e(\tau)} := I_{(v(\tau),w(\tau+\theta_e))} = I_e(\tau)$, and $b_T^{e(\tau)} := b_e(\tau) \ \forall \ e(\tau) \in A_T$. Let $(v(\tau), w(\tau + \theta_e)) \in A_T$ and let $f_e(\tau)$ be a flow in the dynamic network $N = (V, A, u_e(\tau), \theta, T)$. Then the corresponding flow function in $N_T = (V_T, A_T, u_T)$ can be related by

$$f_T^{e(\tau)} := f_{(v(\tau),w(\tau+\theta_e))} = f_e(\tau) \ \text{ for all } e(\tau) \in A_T. \tag{5.18}$$

Relation (5.18) is a bijection from the set of feasible flows in the dynamic network $N = (V, A, u_e(\tau), \theta, T)$ onto the set of feasible flows in $N_T = (V_T, A_T, u_T)$ so that dynamic the flows $f_e(\tau)$ with time horizon T are equivalent to the static flows $f_T^{e(\tau)}$ in $N_T = (V_T, A_T, u_T)$. The flow improvement problem in dynamic network $N = (V, A, u_e(\tau), \theta, T)$ with time horizon T, capacity improvement $I_e(\tau)$ and improvement cost $b_e(\tau)$ is equivalent to the flow improvement problem in static network $N_T =$

(V_T, A_T, u_T) with capacity improvement $I_T^{e(\tau)}$ and improvement cost $b_T^{e(\tau)}$. As the integral maximum flow improvement problem in static network can also be solved optimally (cf. Theorem 5.1), the integral MDFIP can also be solved optimally. With similar arguments, the continuous MDFIP can be solved optimally. □

The dynamic improvement and dynamic improvement cost have been considered in (5.10)-(5.17) model. While constructing time-expanded network in the proof of Theorem 5.5, they are transformed into static improvement and static improvement cost by applying the transformations $I_T^{e(\tau)} := I_{(v(\tau),w(\tau+\theta_e))} = I_e(\tau)$, and $b_T^{e(\tau)} := b_e(\tau) \; \forall \; e(\tau) \in A_T$, respectively. If one consider the improvement and its cost both static in (5.10)-(5.17) model, the time-expanded network copies the arc with the same improved capacity for every arc of the dynamic network. These modifications ensure that each copies of the same arc is improved by the same amount of capacity and the improvement cost of an arc is counted only once for all of its copies.

5.2 Flow Improvements with Arc Switching Costs

Contraflow approach increases the flow value by reversing the directions of arcs towards the sinks as a flow towards the sources is neither preferred nor expected in emergencies. This concept without any reversal cost is firstly incorporated in [130] and analytically studied in [16], where the arc is reversed with its full capacity or left as it is. However, a reversal may require some switching cost. Different contraflow models and solution procedures with switching costs are considered throughout this section.

5.2.1 Maximum static flow problem

The maximum static contraflow problem (MSCFP) introduced in [16] maximizes the source-sink flow value, where directions of arcs have been reversed without considering

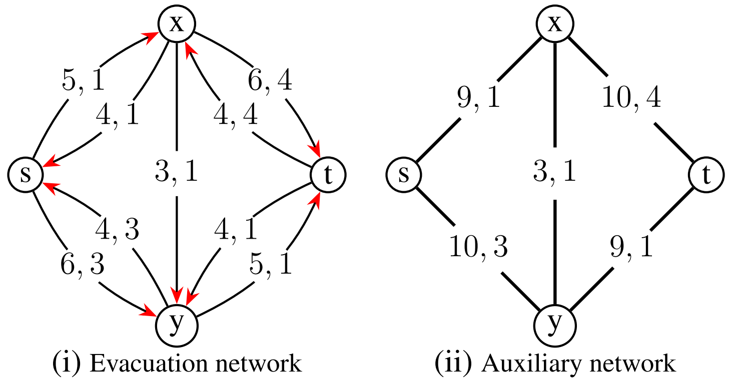


Figure 15: Network with capacities and transit times, respectively.

the reversal costs. In this section, we introduce budget constraint MSCFP and propose its solution procedure for a two-terminal network.

The switching cost contraflow (SCCF) finds a feasible flow in $N = (V, A, u)$, where the arc directions can be reversed accepting switching costs whenever flow can be improved. The static SCCF problem has a similar structure as the minimum concave-cost network flow problem, [158]. These problems ask to find a feasible flow while minimizing the total cost. The total cost is the sum of concave-costs due to the use of arcs by feasible flow. It can be assumed that the concave-cost per arc consists of a fixed cost, whenever this particular arc is used and a variable cost depends on the amount of flow. Fixing the variable cost to zero leads to a special problem called minimum cost fixed flow (MCFF) problem. The improvement strategy function I is a $0/1$ decision if additional capacity is used or not; independent of how much additional capacity is used. An integral flow improvement strategy is considered through out this section.

Problem 5.3. The MSCFP with switching cost is the maximum flow problem in static network where arcs can be reversed accepting some switching costs and total cost is restricted by given budget.

Algorithm 12: Maximum Static Contraflow with Switching Cost

Input : Two-terminal arc reversible network $N = (V, A, u)$ with reversal budget B and reversal cost b_e.

1 Construct an auxiliary network, $\tilde{N} = (V, \tilde{A}, \tilde{u})$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$.

2 Define the cost function by

$$c_{\tilde{e}}(f) = \begin{cases} 0 & \text{for } 0 \leq f_{\tilde{e}} \leq u_{\overrightarrow{e}} \\ b_{\overleftarrow{e}} \dot{(} f_{\tilde{e}} - u_{\overrightarrow{e}}) & \text{for } u_{\overrightarrow{e}} < f_{\tilde{e}} \leq \tilde{u}_{\tilde{e}} \end{cases} \quad \text{for all } \tilde{e} \in \tilde{A}.$$

3 Transform the cost and the capacity functions by

$$\begin{aligned} \overline{u}_{e_0} &:= u_{\overrightarrow{e}} & \quad \overline{u}_{e_1} &:= \tilde{u}_{\tilde{e}} - u_{\overrightarrow{e}} \\ \overline{c}_{e_0} &:= 0 & \quad \overline{c}_{e_1} &:= b_{\overleftarrow{e}}. \end{aligned}$$

4 Find the optimal flow F^*_{st} in auxiliary network considering costs defined in Step 3 and the total budget B, by applying binary search in $[0, n\tilde{U}_{max}]$ where $\tilde{U}_{\max} := \max\{\tilde{u}_{\tilde{e}} : \tilde{e} \in \tilde{A}\}$.

5 An arc $\overleftarrow{e} \in A$ is reversed with capacity $f_{\tilde{e}} - u_{\overrightarrow{e}}$ if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$.

Output: Maximum contraflow value in $N = (V, A, u)$ using arc reversal budget B.

Theorem 5.6. Algorithm 12 solves the Problem 5.3 optimally in polynomial time.

Proof. Step 1 constructs an auxiliary network from the given network which is feasible. Step 2 defines the cost function for the improved capacity which is convex and non-linear whereas Step 3 is to linearize the cost function defined in Step 2. Thus, both these steps are feasible. Step 4 finds the optimal flow F^*_{st} in $\tilde{N} = (V, \tilde{A}, \tilde{u}_{\tilde{e}})$ using binary search method in $[0, n\tilde{U}_{max}]$ and Step 5 reverses the direction of arcs according to the direction of the flow obtained in Step 4. So, Step 4 and Step 5 are also feasible. Hence, the algorithm is feasible.

To restrict on budget, this algorithm only reverses the required capacity of the arc so that $I_e = \max\{0,\ f_{\tilde{e}} - u_{\overrightarrow{e}}\}$ where $f_{\tilde{e}}$ represents the flow through the arc $\tilde{e}$. Cost for the reversed capacity is defined in Step 2 which is not linear but convex and is linearized in Step 3, as in [42, 159]. Then MSCFP with switching cost is equivalent to the budget constraint maximum flow improvement problem with integral improvement which is equivalent to the budget constraint minimum cost flow problem. Thus, the Problem 5.3 is equivalent to budget constraint minimum cost flow problem. Suppose F^*_{st} be the optimal flow achievable on budget B which can be obtained by the method of binary search in $[0, n\tilde{U}_{max}]$.

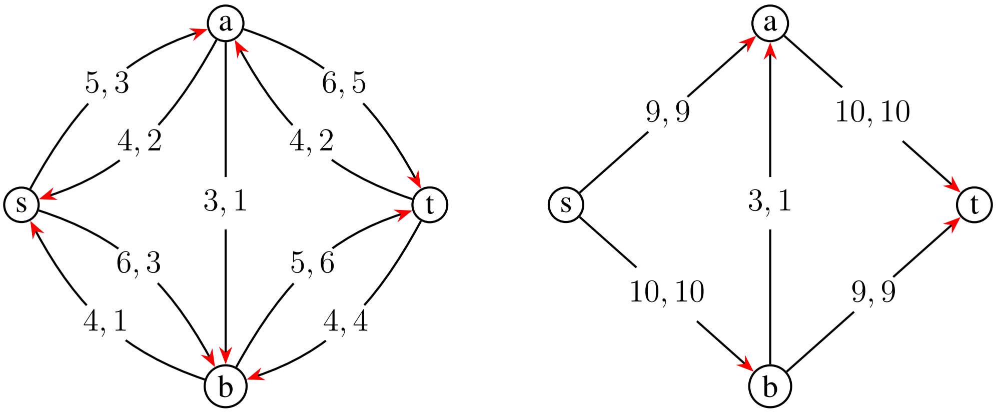
Step 1 constructs an auxiliary network in linear time and is similar for Step 2 and Step 3. Step 5 reverses the arc directions in linear time according to the flow obtained in Step 4 since not both of the arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be switched at a time. Thus, running time of the algorithm depends on Step 4. Hence, Algorithm 12 solves Problem 5.3 in $\mathcal{O}(\log(n\tilde{U}_{max}))$ minimum cost flow computations. □

Algorithm 12 reverses the direction of arcs with integral capacity thus the improvement is integral. This implies that the optimal flow of the problem is integral. For this, the binary search algorithm can test only for the integral flow values of the interval. Instead of applying binary search in the interval, one can apply Megiddo's parametric search, [48] to extract the solution in Step 4 of Algorithm 12.

The maximum value F'_{st} satisfying the budget constraint lies in $\{1, 1+\epsilon, \ldots, (1+\epsilon)^k\}$, where $k = \left\lceil \log_{1+\epsilon}(n\tilde{U}_{max}) \right\rceil$ and $\epsilon > 0$ is a fixed accuracy parameter. This modified binary search finds new value such that $F'_{st} \geq \frac{F^*_{st}}{1+\epsilon}$. If such modification is applied in Step 4 of Algorithm 12, a $(1+\epsilon)$- approximation for budget constrained maximum contraflow can be obtained in $O(\log \log_{1+\epsilon}(n\tilde{U}_{max}))$ minimum cost flow computations where $\epsilon > 0$ is a fixed accuracy parameter.

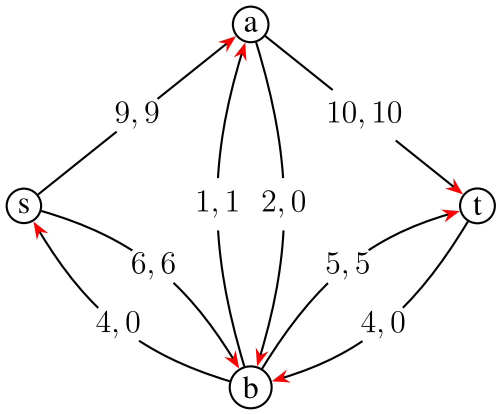
Example 5.1. Let $B = 20$ be the total reversal budget for the evacuation network shown in Figure 16(i). Before contraflow configuration, 10 units of flow could be reached in the sink from the source through the paths $s-a-t$ and $s-b-t$. Likewise, 19 units of flow could be sent through paths $s-a-t$, $s-b-t$ and $s-b-a-t$ (cf. Figure 16(ii)) applying contraflow approach for which $B = 37$ is required for the complete arc reversals which is more than the given budget. This means that the upper bound of the budget constraint contraflow is 37. The budget constraint contraflow can be found applying Algorithm 12.

The optimal budget constraint flow in the auxiliary network Figure 15(i) of Figure 16(i) can be obtained using Step 4 of the algorithm. In this network, $\tilde{U}_{\max} = \max\{\tilde{u}_{\tilde{e}} \mid \tilde{e} \in \tilde{A}\} = \max\{10, 9, 3, 10, 9\} = 10$. Thus, the flow can be obtained in $[0, 4 \times 10] = [0, 40]$ as $n := |V|$. Here, the mid point of the interval is 20 but it is infeasible so that the flow should be searched in the lower interval $[0, 19]$. Again, a middle value of the interval is 10 and its minimum cost is zero as 5 units of flow can be sent through each path $s-a-t$ and $s-b-t$ without reversing any arcs. The objective of the problem is to maximize the flow under the given budget so that flow should be searched in the upper interval $[11, 19]$. The middle value is 15. Its minimum cost is 17 as the reversed arcs are $(a, s), (t, a), (a, b)$ with capacities $4, 4,$ and 1, respectively. Here, we reverse the arc (a, b) only with capacity 1 as the budget should not be wasted. The resulting network flow is shown in Figure 17. Next, the possible upper value should be searched in $[16, 19]$. Let us check the cost for 17. For this, minimum cost is 27 which exceeds the given budget, thus it is infeasible. Similarly, check the minimum cost for 16 units of flow which is 22. Again, this exceeds the budget. Hence, the budget constraint contraflow is 15 which is shown in Figure 17 for which 17 units of budget have been used. Still, 3 units of budget remains unused but it is insufficient.



(i) (capacity, unit cost) (ii) (capacity, flow)

Figure 16: Evacuation and contraflow solution networks without budget constraint, respectively.



(capacity, flow)

Figure 17: Budget constrained contraflow solution of Figure 16(i).

Theorem 5.7. The fixed switching cost contraflow problem is equivalent to $0/1$-MSFIP where all data to be positive integral, [16].

Since the $0/1$ maximum flow improvement is equivalent to maximum fixed cost flow problem, this implies that the FSCCF is $\mathcal{NP}$-hard to solve, Theorem 5.3. The fixed cost for arc reversals makes the problem $\mathcal{NP}$-hard, even in the static case. The static contraflow algorithm is 'blind' for the arc reversal decisions. Adding a time component to FSCCF makes it practically even more difficult to solve. Based on the above results, following theorems are proposed.

Theorem 5.8. The FSCCF problem is $\mathcal{NP}$-hard even in series parallel graphs.

Proof. The FSCCF problem is equivalent to $0/1$-MSFIP, Theorem 5.7. As the maximum flow problem with fixed cost on arcs is equivalent to $0/1$- maximum flow improvement, [42]. Then by Theorem 5.3, it can be claimed that the FSCCF problem is $\mathcal{NP}$-hard even in series parallel graphs. □

Theorem 5.9. There is a fully polynomial-time approximation scheme for the FSCCF problems when the problems are restricted to series parallel graphs.

Proof. As in proof of Theorem 5.8, the FSCCF problem is equivalent to $0/1$-MSFIP and the maximum flow problem with fixed costs on arcs. Since all hardness and approximation results of these problems will be carried over to the FSCCF problem so that a fully polynomial-time approximation scheme can be obtained for the FSCCF problems in series parallel graphs, Theorem 5.4. □

5.2.2 Maximum dynamic flow problem

The notion of MDF is introduced and solved in [16] by reversing the arcs at time zero with zero cost. In this section, reversal cost is considered to maximize the dynamic contraflow subject to budget constraint which is an extension of MSCFP from static to dynamic network.

Problem 5.4. The MDCF problem with switching cost is the MDF problem where arcs can be reversed accepting the switching cost such that the total cost is subject to the budget constraint.

Theorem 5.10. Algorithm 13 solves the Problem 5.4 optimally in pseudo polynomial-time complexity.

Proof. Step 1 is well-defined as it transforms given dynamic network flow problem into the static network flow problem. The feasibility of other steps of the algorithm can be shown as in Theorem 5.6. Thus, the algorithm is feasible.

Algorithm 13: Maximum Dynamic Contraflow with Switching Cost

Input : Two-terminal arc reversible network $N = (V, A, u_e(\tau), \theta, T)$ with reversal budget B, reversal cost $b_e(\tau)$ and symmetric arc capacity.

1 Transform the given dynamic network into the time expanded network by:

$$
\begin{aligned}
V_T &:= \{v(\tau) : v \in V, \tau \in \mathcal{T}\} \\
A_T &:= \{(v(\tau), w(\tau + \theta_e)) \mid e = (v, w) \in A, 0 \leq \tau \leq T - \theta_e\} \\
u_T^{e(\tau)} &:= u_e(\tau), \text{ for } e(\tau) \in A_T \\
b_T^{e(\tau)} &:= b_e(\tau), \text{ for } e(\tau) \in A_T \text{ and}
\end{aligned}
$$

capacities and costs of holdover arcs are infinity and zero, respectively.

2 Construct an auxiliary $\tilde{N}_T = (V_T, \tilde{A}_T, \tilde{u_T})$ as:
$\tilde{e}(\tau) \in \tilde{A}_T$ if $\overrightarrow{e}(\tau) = (v(\tau), w(\tau + \theta_e)) \in A_T$ or
$\overleftarrow{e}(\tau) = (w(\tau - \theta_e), v(\tau)) \in A_T$ with new capacity
$\tilde{u_T}^{\tilde{e}(\tau)} = u_T^{\overrightarrow{e}(\tau)} + u_T^{\overleftarrow{e}(\tau)}$ for all $\tilde{e}(\tau) \in \tilde{A}_T$.

3 Define the cost function by $c_T^{\tilde{e}(\tau)}(f_T^{\tilde{e}(\tau)}) =$

$$
\begin{cases}
0 & \text{for } 0 \leq f_T^{\tilde{e}(\tau)} \leq u_T^{\overrightarrow{e}(\tau)} \\
b_T{}^{\overleftarrow{e}(\tau)} \cdot (f_T^{\tilde{e}(\tau)} - u_T^{\overrightarrow{e}(\tau)}) & \text{for } u_T^{\overrightarrow{e}(\tau)} < f_T^{\tilde{e}(\tau)} \leq \tilde{u}_T^{\tilde{e}(\tau)}
\end{cases}
\quad \forall\, \tilde{e}(\tau) \in \tilde{A}_T.
$$

4 Transform the cost and the capacity functions by

$$
\begin{aligned}
\overline{u}_T^{e_0} &:= u_T^{\overrightarrow{e}(\tau)} & \qquad \overline{u}_T^{e_1} &:= \tilde{u}_T^{\tilde{e}(\tau)} - u_T{}^{\overrightarrow{e}(\tau)} \\
\overline{c}_T^{e_0} &:= 0 & \qquad \overline{c}_T^{e_1} &:= b_T^{\overleftarrow{e}(\tau)}.
\end{aligned}
$$

5 Find the optimal flow in auxiliary network considering budget B and the cost defined in Step 4 using binary search in $[0, nT\tilde{U}_{max}]$, where
$\tilde{U}_{\max} = \max\{\tilde{u}_T^{\tilde{e}(\tau)} : \tilde{e}(\tau) \in \tilde{A}_T\}$.

6 An arc $\overleftarrow{e}(\tau) \in A_T$ is reversed with capacity $\tilde{f_T}^{\tilde{e}(\tau)} - u_T^{\overrightarrow{e}(\tau)}$ if and only if the flow along $\overrightarrow{e}(\tau) \in A_T$ is greater than $u_T^{\overrightarrow{e}(\tau)}$ or there is a non-negative flow along the arc $\overrightarrow{e}(\tau) \notin A_T$.

7 Reconstruct the dynamic network flow from the static network flow according to the flow obtained in Step 5 and arc reversal in Step 6.

Output: Maximum dynamic contraflow in $N = (V, A, u_e(\tau), \theta, T)$ with respect to budget B.

The dynamic network can be transformed into static network as in Step 1. Step 2 constructs the auxiliary network and Step 6 finds an optimal flow in auxiliary network with respect to the cost defined in Step 5 and total budget B. Let $f_e(\tau)$ be the budget constraint contraflow in the dynamic network. Then the corresponding budget constraint flow function in the network $\tilde{N_T} = (V_T, \tilde{A_T}, \tilde{u_T})$ can be related by Equation 5.19.

$$f_T^{\tilde{e}(\tau)} = f_{(v(\tau),w(\tau+\theta_{\tilde{e}}))} = f_e(\tau) \;\; \forall\, \tilde{e}(\tau) \in \tilde{A_T}. \tag{5.19}$$

It is a bijection from the set of feasible budget constraint contraflows in the dynamic network $N = (V, A, u_e(t), \theta, T)$ onto the set of feasible budget constraint flows in $\tilde{N_T} = (V_T, \tilde{A_T}, \tilde{u_T})$ so that the contraflows $f_e(\tau)$ with time horizon T are equivalent to the flows $f_T^{\tilde{e}(\tau)}$ in the time-expanded network. Remaining proof of the theorem can be shown as in Theorem 5.6. Suppose F_{dy}^* be the optimal contraflow achievable on budget B. Hence, the optimal contraflow can be obtained in $\mathcal{O}(\log(Tn\tilde{U}_{max}))$ minimum cost flow computations. $\square$

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

The theoretical backgrounds and basic terminologies for the results presented in the thesis are compactly discussed in Chapter 2, as a basic foundation. The abstract flow and abstract contraflow problem with different objectives in different network structures are given in Chapter 3. The facility location in the evacuation network is presented in Chapter 4. The contraflow problems with switching costs are investigated in Chapter 5.

The abstract flow model deals with the flow paths (routes) that satisfy the switching property. Contraflow increases the flow and decreases the evacuation time making the traffic smooth during evacuation by reversing the required road directions from the risk areas to the safe places. The contraflow approach has been embedded with it to introduce abstract contraflow models to increase the flow values and minimizes the time by reducing crossing conflicts with arc reversals toward the safe destinations in the evacuation network. This model has been extended into abstract contraflow over time and solved efficiently. To maximize the flow at every time point, the earliest arrival abstract contraflow has been solved and is also extended to a multi-terminal network considering supplies and demands. For the given priority of terminals, the lexicographically maximum abstract flow has been introduced and solved polynomially that maximizes the flow respecting the priority.

Appropriate facility locations, contraflow, and transportation facilities are some of the essential components for the efficient evacuation planning flow models. In our work, the network facility location and the contraflow approach are incorporated into the flow models and some efficient algorithms are presented to locate the facility with an objective of minimum flow loss in the evacuation network. Our facility location contraflow solutions obtain optimal plans for the given and arbitrary locations.

The improvement of network topology plays an important role to improve the optimal solution. But, the improvement could be affected by the available incremental budget. Different flow improvement strategies for fixed switching costs are investigated. Different maximum static flow improvement problems have been studied with their complexities and also present for the maximum dynamic flow improvement problem. The contraflow reconfiguration problems with arc switching cost are also introduced concerning the maximum static contraflow problems.

6.2 Conclusions

In our work, we integrate the concepts of abstract flow with the contraflow and provide the mathematical formulations of these models for the first time, and present efficient algorithms for solving the abstract contraflow problems. The efficient solution procedures are presented for maximum dynamic, lexicographically maximum, and earliest arrival abstract contraflow problems. This approach maximizes the flow value in a given time and seeks to reduce the crossing conflicts. However, the abstract contraflow on a multi-terminal dynamic network is $\mathcal{NP}-$Complete. Integrating abstract earliest arrival flow and contraflow, we also introduce the abstract earliest arrival contraflow approach with discrete-time settings on a multi-terminal abstract network for the first time. A 2-value approximate algorithm is proposed for the problem assuming fixed demands and supplies on sources and sinks, respectively. Our results increase the flow values at every possible time by reducing crossing conflicts with arc reversals toward the safe destinations in evacuation planning.

Here, we introduce the ContraFlowLoc model that locates the facility on a given network and maximize the flow in an updated network where arcs can be revered if the optimal solution can be improved from the given network. With a glimpse of the maximum FlowLoc over time, maximum static ContraFlowLoc, and maximum dynamic ContraFlowLoc problems efficient algorithms are also presented for the maximum dynamic FlowLoc, maximum ContraFlowLoc, and maximum dynamic ContraFlowLoc in two-terminal networks for given possible locations as well as extended locations.

It is not an easy task to develop a universally accepted model to handle different issues of evacuation planning with limited resources. To cope with such a situation, the budget-constrained network flow improvement approach plays a significant role. In our work, different flow improvement strategies for fixed switching costs are investigated, namely, integral, rational, and either to increase the full capacity of an arc or not at all. We consider an evacuation problem that aims to shift a maximum number of evacuees from dangerous areas to safe areas in a limited time under the budget constraints for network modification.

In this work, our focus is on the flow improvement problems with switching costs constrained by arc reversal costs on different static and dynamic network flow models. Different variants of flow improvement problems have been studied, and their complexities are given. Here, the contraflow reconfiguration with arc switching costs is introduced. Based on this approach, a polynomial-time algorithm and a pseudo polynomial-time algorithm are proposed for maximum static and maximum dynamic contraflow improvement problems, respectively. The arc capacities in these dynamic flow models are subject to the time-dependent capacity constraints with time-invariant transit times.

6.3 Recommendations for further work

Here the contraflow approach has been embedded in the abstract flow model to introduce and develop different models, algorithms, and the respective solution strategies on abstract contraflow models. However, such a problem on a multi-terminal dynamic network is NP-complete. The facility location can also be incorporated with the contraflow approach for better evacuation planning for what we deal with contraflow model in different aspects. However, its further extensions to achieve different objectives such as lexicographically maximum ContraFlowLoc, lexicographically maximum dynamic ContraFlowLoc, earliest arrival transshipment ContraFlowLoc, and quickest ContraFlowLoc for different networks with both given and arbitrary locations with costs, as the multi-criteria problems are the further research. Different variants of flow improvement strategies for the budget constraints switching costs have been studied with their complexities. By considering different attributes on the arcs further extensions can be done. For example, if both the parameters are taken to be time- or flow-dependent, the hardness increases, and it is our further interest. The findings of these investigations in both theoretical and practical aspects are interesting.

REFERENCES

[1] Ford, L.R. ,& Fulkerson, D.R. (1958). Constructing maximal dynamic flows from static flows. *Operations Research*, **6**(3), 419–433.

[2] Altay, N., & Green III, W.G. (2006). Or/ms research in disaster operations management. *European journal of operational research*, **175**(1), 475–493.

[3] Dhamala, T.N., Dhungana, R.C., & Pyakurel, U. (2017). Dynamic flow models and algorithms for evacuation planning. In International Conference on Computational Modeling and Simulation (ICCMS-2017), 307–311.

[4] Dhamala, T.N., Pyakurel, U., & Dempe, S. (2018). A critical survey on the network optimization algorithms for evacuation planning problems. *International Journal of Operations Research*, **15**(3), 101–133.

[5] Hamacher, H.W., & Tjandra, S.A. (2001). Mathematical modeling of evacuation problems: a state of the art, In Pedestrian and Evacuation Dynamics. (eds. M. Schreckenberger and S.D. Sharma), Springer.

[6] Jarvis, J.J., & Ratliff, H.D. (1982). Some equivalent objectives for dynamic network flow problems. *Management Science*, **28**(1), 106–109.

[7] Moriarty, K.D., Ni, D., & Collura, J. (2007). Modeling traffic flow under emergency evacuation situations: Current practice and future directions. In Transportation Research Board 2007 Annual Meeting.

[8] Pel, A.J., Bliemer, M.C., & Hoogendoorn, S.P. (2012). A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation*, **39**(1), 97–123.

[9] Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., & Seyfried, A. (2009). *Evacuation dynamics: Empirical results, modeling and applications*. Springer, New York.

[10] Hamacher, H.W., & Tufekci, S. (1987). On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*, **34**, 487–503.

[11] Ford, L.R. & Fulkerson, D.R. *Flows in networks*. Princeton University Press, Princeton, New Jersey.

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

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

[14] 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.

[15] Arulselvan, A. (2009). *Network model for disaster management*. PhD thesis, University of Florida Gainesville.

[16] 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.

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

[18] Dhungana, R.C., Pyakurel, U., Khadka, S., & Dhamala, T.N. (2015). Universally maximum contraflow for evacuation planning. *International Journal of operations research*, **4**, 67–78.

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

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

[21] Pyakurel, U., & Dhamala, T.N. (2017). Continuous dynamic contraflow approach for evacuation planning. *Annals of Operations Research*, **253**(1), 573–598.

[22] Pyakurel, U., & Dhamala, T.N. (2017). Evacuation planning by earliest arrival contraflow. *Journal of Industrial & Management Optimization*, **13**(1), 489–503.

[23] Pyakurel, U., Hamacher, H.W., & Dhamala, T.N. (2014). Generalized maximum dynamic contraflow on lossy network. *International Journal of Operations Research Nepal*, **3**(1), 27–44.

[24] Pyakurel, U. (2016). *Evacuation planning problem with contraflow approach*. PhD thesis, IOST, Tribhuvan University, Nepal.

[25] Pyakurel, U., Nath, H.N., Dempe, S., & Dhamala, T.N. (2019). Efficient dynamic flow algorithms for evacuation planning problems with partial lane reversal. *Mathematics*, **7**(10), 993.

[26] Dhungana, R.C., Pyakurel, U., & Dhamala, T.N. (2018). Abstract contraflow models and solution procedures for evacuation planning. *Journal of Mathematics Research*, **10**(4), 89–100.

[27] Dhungana, R.C., & Dhamala, T.N. (2019). Maximum flowloc problems with network reconfiguration. *International Journal of Operations Research*, **16**(1), 13–26.

[28] Pyakurel, U., Nath, H.N., & Dhamala, T.N. (2019). Partial contraflow with path reversals for evacuation planning. *Annals of Operations Research*, **283**(1-2), 591–612.

[29] Mo, J., Gao, M., & Liu, L. (2019). An improved critical-edge model for finding optimal contraflow links considering the influence of intersections. *Mathematical Problems in Engineering*, **2019**.

[30] Dhungana, R.C., & Dhamala, T.N. (2018). Abstract earliest arrival transshipment with network reconfiguration. The Nepali Mathematical Science Report, **35**(1-2), 1-9.

[31] Kappmeier, J.-P.W. (2015). *Generalizations of flows over time with application in evacuation optization*. PhD thesis, Technical University, Berlin, Germany.

[32] Kappmeier, J.-P.W., Matuschke, J., & Peis, B. (2014). Abstract flows over time: A first step towards solving dynamic packing problems. *Theoretical Computer Science*, **544**, 74–83.

[33] Dhungana, R.C., & Dhamala, T.N. (2018). Flowloc problems on evacuation network. In The 11th Triennial Conference of Association of Asia Pacific Operational Research Societies (APORS 2018, August 6-9), 121–123. Operational Research Society, Nepal.

[34] Dhungana, R.C., & Dhamala, T.N. (2017). Flow improvement with fixed switching cost in network flow models. In National Conference on Mathematics and Its Applications (NCMA-2017), 85–90. Nepal Mathematical Society.

[35] Dhungana, R.C., & Dhamala, T.N. (2020). Flow improvement in evacuation planning with budget constrained switching costs. *International Journal of Mathematics and Mathematical Sciences*, Article ID:1605806.

[36] Hoffman, A. (1974). A generalization of max flow-min cut. *Mathematical Programming*, **6**(1), 352–359.

[37] Zhao, X., Feng, Z.-Y. Li., Y., & Bernard, A. (2016). Evacuation network optimization model with lane-based reversal and routing. *Mathematical Problems in Engineering*, **2016**.

[38] Weber, A. (1929). Über den Standort der Industrien, Tübingen. *Theory of the Location of Industries*. University of Chicago Press (English translation by C. J. Friedrich).

[39] Daskin, M.S. (2011). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons.

[40] Drezner, Z., & Hamacher, H.W. (2004). *Facility location: Applications and theory*. Springer Science & Business Media.

[41] Hamacher, H.W., Heller, S., & Rupp, B. (2013). Flow location (flowloc) problems: dynamic network flows and location models for evacuation planning. *Annals of Operations Research*, **207**(1), 161–180.

[42] Krumke, S.O., Noltemeier,H., Schwarz, S., Wirth, H.-C., & Ravi, R.(2009). Flow improvement and network flows with fixed costs. In Operations Research Proceedings 1998, Springer, 158–167.

[43] Ahuja, R.K., Magnanti, T.L., Orlin,J. B., & Weihe, K. (1995). Network flows: theory, algorithms and applications. *ZOR-Methods and Models of Operations Research*, **41**(3), 252–254.

[44] Bang-Jensen, J., & Gutin, G.Z. (2008). *Digraphs: theory, algorithms and applications*. Springer Science & Business Media.

[45] Grötschel, M., Lovász, L., & Schrijver, A. (2012).*Geometric algorithms and combinatorial optimization*, Springer Science & Business Media.

[46] Papadimitriou, C.H., & Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

[47] Williamson, D.P., & and Shmoys, D.B. (2011). *The design of approximation algorithms*. Cambridge University Press.

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

[49] Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**(1), 269–271.

[50] Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, **16**(1), 87–90.

[51] Ford, L.R. (1956). Network flow theory. Technical report, Rand Corp Santa Monica CA.

[52] Moore, E.F. (1959). The shortest path through a maze. In Proceedings of an International Symposium on the Theory of Switching 1957, Part II, Harvard University Press, Cambridge, USA, 285–292.

[53] Harris, T.E., & Ross, F.S. (1955). Fundamentals of a method for evaluating rail net capacities. Technical report, The RAND Corporation Research Memorandum RM–1573.

[54] Ford, L.R. & Fulkerson, D.R. (1956). Maximal flow through a network. *Canadian Mathematical Society*, **8**, 399–404.

[55] Elias, P., Feinstein, A., & Shannon, C. (1956). A note on the maximum flow through a network. *IRE Transactions on Information Theory*, **2**(4), 117–119.

[56] Schrijver, A. (1984). Total dual integrality from directed graphs, crossing families, and suband supermodular functions. In Progress in Combinatorial Optimization, Elsevier, 315–361.

[57] Dantzig, G., & Fulkerson, D.R. On the max flow min cut theorem of networks. *Linear Inequalities and Related Systems Annals of Mathematics Studies*, **38**, 225–231.

[58] Ford, L.R. & Fulkerson, D.R. (1957). A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian Journal of Mathematics*, **9**, 210–218.

[59] Boldyreff, A.W. (1955). Determination of the maximal steady state flow of traffic through a railroad network. *Journal of the Operations Research Society of America*, **3**(4), 443–465.

[60] Schrijver, A. (2002). On the history of the transportation and maximum flow problems. *Mathematical Programming*, **91**(3), 437–445.

[61] Zwick, U. (1995). The smallest networks on which the ford-fulkerson maximum flow procedure may fail to terminate. *Theoretical Computer Science*, **148**(1), 165–170.

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

[63] Dinic, E.A. (1970). Algorithm for solution of a problem of maximum flow in networks with power estimation. In Soviet Math. Doklady, **11**, 1277–1280.

[64] Karzanov, A.V. (1974). Determining the maximal flow in a network by the method of preflows. In Soviet Math. Doklady, **15**, 434–437.

[65] Malhotra, J.V., Kumar, M.P., & Maheshwari, S.N. (1978). An $\mathcal{O}(|V|^3)$ algorithm for finding maximum flows in networks. *Information Processing Letters*, **7**(6):277–278.

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

[67] King, V., Rao, S., & Tarjan, R. (1994). A faster deterministic maximum flow algorithm. *Journal of Algorithms*, **17**(3), 447–474.

[68] Goldberg, A.V., & Rao, S. (1988). Beyond the flow decomposition barrier. *Journal of the ACM (JACM)*, **45**(5), 783–797.

[69] Cherkassky, B.V., & Goldberg, A.V. (1997). On implementing the push—relabel method for the maximum flow problem. *Algorithmica*, **19**(4), 390–410.

[70] Goldberg, A.V. (2008). The partial augment–relabel algorithm for the maximum flow problem. In European Symposium on Algorithms, Springer, 466–477.

[71] Orlin, J.B. (2013). Max flows in $\mathcal{O}(nm)$ time, or better. In Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, 765–774.

[72] McCormick, S.T. (1996). A polynomial algorithm for abstract maximum flow. In Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, Citeseer, 490–497.

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

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

[75] Gallo, G., Grigoriadis, M.D., & Tarjan, R.E. (1989). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, **18**(1), 30–55.

[76] Hoppe, B., & Tardos, É. The quickest transshipment problem. *Mathematics of Operations Research*, **25**(1), 36–62.

[77] Hoppe, B., & Tardos, É. (1994). Polynomial time algorithms for some evacuation problems. In Annual ACM-SIAM symposium on Discrete Algorithms, **5**, 433–441.

[78] Garg, N., & Köenemann, J. (2007). Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, **37**(2), 630–652.

[79] Vygen, J. (2004). Near-optimum global routing with coupling, delay bounds, and power consumption. In International Conference on Integer Programming and Combinatorial Optimization, Springer, 308–324.

[80] Ahuja, R.K., Mangati, T.L., & Orlin, J.B. (1993). *Network flows theory, algorithms and applications*. Prentice Hall, Englewood Cliffs, New Jersey.

[81] Zadeh, N. (1973). A bad network problem for the simplex method and other minimum cost flow algorithms. *Mathematical Programming*, **5**(1), 255–266.

[82] Tomizawa, N. (1971). On some techniques useful for solution of transportation network problems. *Networks*, **1**(2), 173–194.

[83] Orlin, J.B. (1993). A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, **41**(2), 338–350.

[84] Cunningham, W.H. (1976). A network simplex method. *Mathematical Programming*, **11**(1), 105–116.

[85] Dantzig, G.B. (1951). Application of the simplex method to a transportation problem. Activity analysis and production and allocation.

[86] Goldberg, A.V. (1997). An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, **22**(1), 1–29.

[87] Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, **26**(9), 1124–1137.

[88] Eppstein, D., Galil, Z., & Italiano, G.F. (1998). Dynamic graph algorithms. *Algorithms and Theory of Computation Handbook*, **1**(8), 18–25.

[89] Philpott, A.B. (1982). *Algorithms for continuous network flow problems*. PhD thesis, University of Cambridge.

[90] Melkonian, V. (2007). Flows in dynamic networks with aggregate arc capacities. *Information Processing Letters*, **101**(1), 30–35.

[91] Dressler, D. & Skutella, M. (2010). An fptas for flows over time with aggregate arc capacities. In International Workshop on Approximation and Online Algorithms, Springer, 106–117.

[92] Harks, T., König, F.G., Matuschke, J. Richter, A.T., & Schulz, J. (2016). An integrated approach to tactical transportation planning in logistics networks. *Transportation Science*, **50**(2), 439–460.

[93] Köhler, E., Langkau, K., & Skutella, M. (2002). Time-expanded graphs for flow-dependent transit times. In European Symposium on Algorithms, Springer, 599–611.

[94] Köhler, E., Möhring, R.H., & Skutella, M. (2009). Traffic networks and flows over time. In Algorithmics of Large and Complex Networks, Springer, 166–196.

[95] Han, Y., Guan, X., & Shi, L. (2011). Optimization based method for supply location selection and routing in large-scale emergency material delivery. *IEEE Transactions on Automation Science and Engineering*, **8**(4), 683–693.

[96] Peis, B., & Wiese, A. (2011). Universal packet routing with arbitrary bandwidths and transit times. In International Conference on Integer Programming and Combinatorial Optimization, Springer, 362–375.

[97] Andreatta, G., De Giovanni, L., & Salmaso, G. (2010). Fleet quickest routing on grids: A polynomial algorithm. *International Journal of Pure and Applied Mathematics*, **62**(4), 419–432.

[98] Roy, K., & Tomlin, C.J. (2007). Solving the aircraft routing problem using network flow algorithms. In 2007 American Control Conference, IEEE, 3330–3335.

[99] Fulkerson, D.R. (1961). A network flow computation for project cost curves. *Management Science*, **7**(2), 167–178.

[100] Wang, J., Wang, H., Zhang, W., Ip, W., & Furuta, K. (2013). Evacuation planning based on the contraflow technique with consideration of evacuation priorities and traffic setup time. *IEEE Transactions on Intelligent Transportation Systems*, **14**(1), 480–485.

[101] Soldati, P., Zhang, H., & Johansson, M. (2009). Deadline-constrained transmission scheduling and data evacuation in wirelesshart networks. In European Control Conference (ECC), IEEE, 4320–4325.

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

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

[104] Koch, R., & Skutella, M. (2011). Nash equilibria and the price of anarchy for flows over time. *Theory of Computing Systems*, **49**(1), 71–97.

[105] Fonoberova, M.A., & Lozovanu, D.D. (2004). The maximum flow in dynamic networks. *Computer Science Journal of Moldova*, **12**(3), 36.

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

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

[108] Hall, A., Hippler, S. & Skutella, M. (2007). Multicommodity flows over time: Efficient algorithms and complexity. *Theoretical Computer Science*, **379**(3), 387–404.

[109] Groß, M., & Skutella, M. (2012). Maximum multicommodity flows over time without intermediate storage. In European Symposium on Algorithms, Springer, 539–550.

[110] Groß, M. (2009). *Flows over time with commodity-dependent transit times*. Master's thesis, Technische Universität Dortmund.

[111] Klinz, B., & Woeginger, G.J. (1995). Minimum cost dynamic flows: The series-parallel case. In International Conference on Integer Programming and Combinatorial Optimization, Springer, 329–343.

[112] Klinz, B., & Woeginger, G.J. (2004). Minimum-cost dynamic flows: The series-parallel case. *Networks: An International Journal*, **43**(3), 153–162.

[113] Orlin, J.B. (1984). Minimum convex cost dynamic network flows. *Mathematics of Operations Research*, **9**(2), 190–207.

[114] Cai, X., Sha, D. & Wong, C. (2001). Time-varying minimum cost flow problems. *European Journal of Operational Research*, **131**(2), 352–374.

[115] Chalmet, L.G., Francis, R.L., & Saunders, P.B. (1982). Network models for building evacuation. *Fire Technology*, **18**(1), 90–113.

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

[117] Lin, M., & Jaillet, P. (2014). On the quickest flow problem in dynamic networks–a parametric min-cost flow approach. In Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 1343–1356.

[118] Saho, M., & Shigeno, M. (2017). Cancel-and-tighten algorithm for quickest flow problems. *Networks*, **69**(2), 179–188.

[119] Fleischer, L. & Skutella, M. (2002). The quickest multicommodity flow problem. In International Conference on Integer Programming and Combinatorial Optimization, Springer, 36–53.

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

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

[122] Tjandra, S.A. (2003). *Dynamic network optimization with applications to evacuation planning*. PhD thesis, University of Kaiserslautern.

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

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

[125] Baumann, N., & Köhler, E. (2007). Approximating earliest arrival flows with flow-dependent transit times. *Discrete Applied Mathematics*, **155**(2), 161–171.

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

[127] Hajek, B., & Ogier, R.G. (1984). Optimal dynamic routing in communication networks with continuous traffic. *Networks*, **14**(3), 457–487.

[128] Hamza-Lup, G.L., Hua, K.A., Lee, M., & Peng, R. (2004). Enhancing intelligent transportation systems to improve and support homeland security. In The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749), 250–255.

[129] Litman, L. (2006). Lessons from katrina and rita: What major disasters can teach transportation planners. *Journal of Transportation Engineering*, **132**(1), 11–18.

[130] Kim, S., & Shekhar, S. (2005). Contraflow network reconfiguration for evacuation planning: a summary of results. In Proceedings of the 13th annual ACM international workshop on Geographic information systems, 250–259.

[131] Wang, J., Ip, W., & Zhang, W.-J.(2010). An integrated road construction and resource planning approach to the evacuation of victims from single source to multiple destinations. *IEEE Transactions on Intelligent Transportation Systems*, **11**(2), 277–289.

[132] Xiong, H., Chen, M., Lin, Y., Lv, N., Yan, X., Xu, K., & Wu. C. (2010). Bi-level programming based contra flow optimization for evacuation events. Kybernetes.

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

[134] Dhamala, T.N., Gupta, S.P., Khanal, D.P., & Pyakurel, U. (2020). Quickest multicommodity flow over time with partial lane reversals. *Journal of Mathematics and Statistics*, DOI: 10.3844/jmssp.2020.198.211.

[135] Dhungana, R.C. (2014). *Efficient algorithm on universally maximum contraflow*. Master thesis, Central Department of Mathematics, Tribhuvan University.

[136] Pyakurel, U., & Dhamala, T.N. (2014). Earliest arrival contraflow model for evacuation planning. *Neural, Parallel, and Scientific Computations*, **22**, 287–294.

[137] 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.

[138] Gerasimenko, E. & Kureichik, V. (2020). The maximum lexicographic contraflow finding in a fuzzy dynamic network. In International Conference on Intelligent and Fuzzy Systems, Springer, 981–989.

[139] Dhamala, T.N., & Adhikari, I.M. (2018). On evacuation planning optimization problems from transit-based perspective. *International Journal of Operations Research*, **15**(1), 29–47.

[140] Bish, D.R. (2011). Planning for a bus-based evacuation. *OR Spectrum*, **33**(3), 629–654.

[141] Goerigk, M., Grün, B., & Heßler, P. (2013). Branch and bound algorithms for the bus evacuation problem. *Computers & Operations Research*, **40**(12), 3010–3020.

[142] Goerigk, M., & Grün, B. (2014). A robust bus evacuation model with delayed scenario information. *OR Spectrum*, **36**(4), 923–948.

[143] Pyakurel, U., Goerigk, M., Dhamala, T.N., & Hamacher, H.W. (2016). Transit dependent evacuation planning for kathmandu valley: A case study. *International Journal of Operations Research Nepal*, **5**, 49–73.

[144] Adhikari, I.M., Pyakurel U., & Dhamala T.N. (2020). An integrated solution approach for the time minimization evacuation planning problem. *International Journal of Operation Research*, **17**(1), 25–37.

[145] Adhikari, I.M., & Dhamala, T.N. (2020). Minimum clearance time on the prioritized integrated evacuation network. *American Journal of Applied Mathematics*, **8**(4), 207–215.

[146] Anaya-Arenas, A.M., Renaud, J., & Ruiz, A. (2014). Relief distribution networks: a systematic review. *Annals of Operations Research*, **223**(1), 53–79.

[147] Colson, B., Marcotte, P. & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, **153**(1), 235–256.

[148] Lim, G.J., Baharnemati, M.R., & Kim, S.J. (2016). An optimization approach for real time evacuation reroute planning. *Annals of Operations Research*, **238**(1-2), 375–388.

[149] Akbari, V., & Salman, F.S. (2017). Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. *European Journal of Operational Research*, **257**(2), 625–640.

[150] Xie, C., & Turnquist, M.A. (2011). Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. *Transportation Research Part C: Emerging Technologies*, **19**(1), 40–63.

[151] Xie, C., Lin, D.-Y., & Waller, S.T. (2010). A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E: Logistics and Transportation Review*, **46**(3), 295–316.

[152] Hua, J., Ren, G., Cheng, Y., & Ran, B. (2014). An integrated contraflow strategy for multimodal evacuation. *Mathematical Problems in Engineering*, **2014**.

[153] Grötschel, M., Lovász, L., & Schrijver, A. (1988). Geometric algorithms and combinatorial optimization. *Algorithms and Combinatorics*, **2**, 65–84.

[154] Martens, M., & McCormick, S.T. (2008). A polynomial algorithm for weighted abstract flow. In International Conference on Integer Programming and Combinatorial Optimization, Springer, 97–111.

[155] Nath, H.N., & Dhamala, T.N. (2018). Network flow approach for locating optimal sink in evacuation planning. *International Journal of Operations Research*, **15**(4), 175–185.

[156] Heller, S., & Hamacher, H.W. (2011). The multi terminal q-flowloc problem: A heuristic. In International Conference on Network Optimization, Springer, 523–528.

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

[158] Guisewite, G.M., & Pardalos, P.M. (1990). Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, **25**(1), 75–99.

[159] Schwarz, S., & Krumke, S.O. (1998). On budget-constrained flow improvement. *Information Processing Letters*, **66**(6), 291–297.

APPENDIX A

LIST OF ALGORITHMS

APPENDIX B

INDEX

APPENDIX C

LIST OF PRESENTATIONS

1. Paper: Contraflow problems with fixed switching costs. The Second International Conference on Applications of Mathematics to Nonlinear Sciences (AMNS-2019), Pokhara, Nepal, June 27–30, 2019.

2. Paper: Abstract FlowLoc problems for evacuation planning. Recent Advances in Informatics, Communication, Management, Health & Applied Sciences (RAICMHAS-2019), Brainware University, Kolkata, India, February, 2–4, 2019.

3. Paper: FlowLoc problems in evacuation network. The 11th Triennial Conference of Association of Asia Pacific Operational Research Societies (APORS-2018), Kathmandu, Nepal, August 6-9, 2018.

4. Paper: Abstract contraflow for evacuation planning. Workshop for UK-Nepal Co-operation in Emergency Management Research, Jointly organized by: Lancaster University, United Kingdom and Central Department of Mathematics, Tribhuvan University, Lalitpur, Nepal, September 1–2, 2017.

5. Paper: Lane based generalized flows in lossy network for evacuation planning. National Conference on History and Recent Trends of Mathematics, Balmeeki Campus, Nepal Sanskrit University (NSU), Kathmandu, Nepal, June 2–4, 2017.

6. Paper: Network flow models and their solution procedures for evacuation planning. Workshop on Bi-level Optimization, Central Department of Mathematics, Kirtipur, Kathmandu, 28 February–7 March, 2017.

7. Paper: Flow improvement with fixed switching cost in network flow models. National Conference on Mathematics and Its Applications (NCMA-2017), Birendra Multiple Campus, Bharatpur, Chitwan, Nepal, January 11–13, 2017.

8. Paper: Significants of contraflow. Sudwest-Workshop "Operations Research", TU Kaiserslautern/Fraunhofer ITWM, Germany, 28 October, 2016.

9. Poster: Evacuation planning problem using abstract flow. The First International Conference on Applications of Mathematics to Nonlinear Sciences (AMNS-2016), Kathmandu, Nepal, May 26–29, 2016.

10. Poster: Evacuation through universally maximum contraflow. Living under Threat of Earthquake, International Conference Humboldt Kolleg, Kathmandu, Nepal, February 19–22, 2016.

11. Paper: Universally quickest contraflow problem in grid network. Two Days Seminar on Recent Developments in Mathematics and its Application, Central Department of Mathematics, TU, Nepal, September 22–23, 2015.

APPENDIX D

LIST OF PUBLICATIONS

1. Dhungana, R. C. and Dhamala, T. N. (2020). Flow improvement in evacuation planning with budget constrained switching costs. International Journal of Mathematics and Mathematical Sciences, Article ID:1605806.

2. Dhungana, R. C. and Dhamala, T. N. (2019). Maximum FlowLoc problems with network reconfiguration. International Journal of Operations Research, 16(1), 13–26.

3. Dhungana, R. C., Pyakurel, U., and Dhamala, T .N. (2018). Abstract contraflow models and solution procedures for evacuation planning. Journal of Mathematics Research, 10(4), 89–100.

4. Dhungana, R. C. and Dhamala, T. N.(2018). Abstract earliest arrival transshipment with network reconfiguration. The Nepali Mathematical Science Report, 35(1-2), 1–9.

5. Dhungana, R. C. and Dhamala, T. N. (2018). FlowLoc problems on evacuation network. The 11th Triennial Conference of Association of Asia Pacific Operational Research Societies (APORS 2018, August 6-9), 121–123.

6. Dhamala, T .N., Dhungana, R. C., and Pyakurel, U. (2017). Dynamic flow models and algorithms for evacuation planning. Proceeding on International Conference on Computational Modeling and Simulation (ICCMS-2017), 307-311.

7. Dhungana, R.C. and Dhamala,T. N. (2017), Flow improvement with fixed switching cost in network flow models. Proceeding on National Conference on Mathematics and Its Applications (NCMA-2017), 85-90.

Hindawi
International Journal of Mathematics and Mathematical Sciences
Volume 2020, Article ID 1605806, 10 pages
https://doi.org/10.1155/2020/1605806

Research Article

Flow Improvement in Evacuation Planning with Budget Constrained Switching Costs

Ram Chandra Dhungana and Tanka Nath Dhamala

Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

Correspondence should be addressed to Tanka Nath Dhamala; amb.dhamala@daadindia.org

Received 9 January 2020; Revised 8 April 2020; Accepted 30 April 2020; Published 1 June 2020

Academic Editor: Frank Werner

Many large-scale natural and human-created disasters have drawn the attention of researchers towards the solutions of evacuation planning problems and their applications. The main focus of these solution strategies is to protect the life, property, and their surroundings during the disasters. With limited resources, it is not an easy task to develop a universally accepted model to handle such issues. Among them, the budget-constrained network flow improvement approach plays significant role to evacuate the maximum number of people within the given time horizon. In this paper, we consider an evacuation planning problem that aims to shift a maximum number of evacuees from a danger area to a safe zone in limited time under the budget constraints for network modification. Different flow improvement strategies with respect to fixed switching cost will be investigated, namely, integral, rational, and either to increase the full capacity of an arc or not at all. A solution technique on static network is extended to the dynamic one. Moreover, we introduce the static and dynamic maximum flow problems with lane reversal strategy and also propose efficient algorithms for their solutions. Here, the contraflow approach reverses the direction of arcs with respect to the lane reversal costs to increase the flow value. As an implementation of an evacuation plan may demand a large cost, the solutions proposed here with budget constrained problems play important role in practice.

1. Introduction

People are living under the threats of different natural and human-created disasters, such as hurricanes, floods, wildfires, or chemical spills. Many disasters are uncertain and unavoidable, but their effects can be minimized with efficient evacuation planning. But, the development of efficient models and algorithms for these planning problems is always challenging. To find efficient transportation routes during the evacuation, network flow models have been widely used. An evacuation network is interpreted by using a directed graph where the intersections of roads are represented by nodes, road segments between nodes are represented by arcs, and routes between two nodes are taken as directed paths. The places where evacuees are gathered and start to move at risk are considered as source nodes, and the safe destinations where they are supposed to arrive are sink nodes. Each node has a nonnegative integer capacity which bounds the maximum possible flow amount through it. Every arc has a cost or a transit time assigned to it. A flow in the network is considered as the evacuees or the vehicles carrying evacuees.

The network flow problems are to find maximum flow or minimum cost flow in a given network. The maximum dynamic flow problem (MDF) in two-terminal networks is solved polynomially using a static minimum cost flow solution [1]. A flow maximization seeks to send as much flow amount as possible within a time bound. A large number of researchers have studied different flow models for various objectives such as the earliest arrival flow to maximize the flow at every possible time; quickest flow to shift given amount of flow in minimum time; lexicographically maximum flow to maximize the flow in given priority order; and quickest transshipment problem to satisfy given demand and supply in minimum time. These dynamic flow models have widely been used in solving several evacuation planning problems. Static flow solutions are the building blocks for dynamic flow solutions. Usually, the evacuation plans should respect the given time bound that may be continuous or discrete. The authors in [2] show that approximate continuous time solution can be obtained by applying the natural transformation to a discretized time solution. Most of these models, except that of cost minimization problems themselves, consider the travel time on an arc as only the cost on it and do not take care of any additional costs occurred during the evacuation plans. They aim to fulfill the respective objectives on fixed network topology. For details, we refer to [3] and [4] and the references therein.

Different types of network modification problems exist in the literature. Generally, the original network is assumed to be not modifiable in the sense that capabilities or costs remain fixed as in the given network. However, this assumption is not valid in many real evacuation scenarios. For example, the capacity of an arc can also be increased up to some limit subject to some capacity incremental cost. For this, a fixed budget can be distributed to increase capacities in the network such that the network topology is modified and an objective, for instance, the flow, with respect to new capacities is maximized. There are three variants of this improvement strategy that deal with rational, integral, and either of the all possible or not at all capacity values in [5]. The first two variants are polynomial time solvable, while the last one is $\mathcal{NP}$-hard even in the cases of bipartite and series-parallel graphs. This third variant called the 0/1 maximum flow improvement strategy is equivalent to the maximum flow-fixed cost problem which is a bicriteria optimization problem where the flow has to be maximized under the budget constraint. A fully polynomial time approximation scheme for series-parallel graphs is presented. The network modification problems that relate to arc-based improvement and node-based upgrading models are also investigated.

Contraflow increases the outbound road capacities by reversing the direction of arcs towards the safe destinations. It increases the flow value and decreases the evacuation time by reducing the congestion in an emergency or rush-hour traffic management. The arc reversals are performed on the existing networks with permissible lane reversals without any additional costs. The authors in [6, 7] prove that the contraflow problems for general networks are $\mathcal{NP}$-hard. The former have presented different heuristics for multi-terminal network, and the latter have polynomially solved maximum dynamic contraflow (MDCF) and quickest contraflow problems in the case of two-terminal networks, respectively. By introducing the contraflow approach, different evacuation planning problems are efficiently solved in [8, 9]. The earliest arrival transshipment contraflow in multisource networks and with zero transit time in multisink networks has been polynomially solved. The authors in [10] investigate the quickest contraflow problem with constant- and load-dependent transit times. The authors in [11] have developed a class of contraflow algorithms and performed computational experiments. The technique of lane reversals is beneficial for other purposes, for example, crossing eliminations, logistic supports, and use of emergency vehicles and facility locations. The contraflow approach with crossing elimination, facility location-allocation, and partial lane reversal strategies are introduced in [12–14], respectively. The third approach makes use of nonreversed arcs in contraflow for supporting facilities and emergency logistics. Multimodel integrated contraflow for uncertain arrivals of evacuees in an evacuation region with a low mobility population is presented in [15]. Considering the influence of intersections, an improved critical-road model has been investigated to find the optimal contraflow links [16].

For provided limited resources, it is not possible to select all arc reversals as demanded by the optimal contraflows. In this paper, we investigate contraflow problems with fixed budget constraint distributed to the arc reversals. The total given budget allows us to reverse only a certain percentage of arcs in a given network. We introduce the maximum dynamic flow improvement problem (MDFIP) and also the maximum contraflow improvement problems in both static and dynamic networks. Then, we propose polynomial time algorithms to solve these problems. To the best of our knowledge, this is the first attempt to incorporate the issues of arc reversal costs on contraflow problems subject to the given total budget constraint. As arc reversals require a lot of costs at emergency period, this approach is more practicable in implementing the contraflow algorithms.

The paper is organized as follows. The network flow models are given in Section 2. The solutions on static and dynamic flow improvement problems are presented in Sections 3.1 and 3.2, respectively. The contraflow models and their solution procedures with unit switching costs on arcs for the flow improvements are proposed in Section 4. Section 5 concludes the paper.

2. Preliminaries

Consider a directed dynamic network $N = (V, A, u_e(t), \theta_e, T)$, with $V = V^+ \cup V^- \cup V^0$, where V^+, V^-, and V^0 represent sets of sources, sinks, and intermediate nodes, respectively. In particular, $V^+ = \{s\}$ and $V^- = \{t\}$, for any two-terminal network. The egress time T is denoted by $\mathcal{T} = \{0, 1, \ldots, T\}$ and $\mathbf{T}_c = \{[0, 1), \ldots, [T, T+1)\}$ in discrete and continuous time models, respectively. The functions θ: $A \longrightarrow R_{\geq 0}$ and u: $A \times \mathcal{T} \longrightarrow R_{\geq 0}$ are arc transit time and capacity functions, respectively. The capacity function bounds the amount of flow on arcs. Suppose f: $A \times \mathcal{T} \longrightarrow R_{\geq 0}$ is a dynamic flow function, where the value $f_e(t)$ entering the arc $e = (v, w)$ at time t also arrives at node w at $t + \theta_e$. The flow excesses at the node induced by a flow on arcs is denoted by h: $V \times \mathcal{T} \longrightarrow R_{\geq 0}$. A directed static network $N = (V, A, u_e)$ is obtained by omitting the time components. For the sake of simplicity, the same functions f and h will be used for static and dynamic flows.

2.1. Network Flow Models

2.1.1. Maximum Static Flow Model. The maximum static flow model is to maximize Objective (1) satisfying Constraints (2)–(4):

$$\max \sum_{v \in V^+} h_v, \quad (1)$$

$$\sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e = \begin{cases} h_v, & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \\ -h_v, & \text{if } v \in V^-, \end{cases} \quad (2)$$

$$h_v \geq 0, \quad \forall v \in V, \quad (3)$$

$$u_e \geq f_e \geq 0, \quad \forall e \in A, \quad (4)$$

where $A_v^+ = \{(v, w) \in A \mid w \in V\}$ for $v \in V$ and $A_v^- = \{(u, v) \in A \mid u \in V\}$. Outgoing flow from the sources, conservation of flow in intermediate nodes, and entering flow to the sinks are, respectively, shown in Constraint (2). Constraint (4) represents bounds of flows on arcs.

2.1.2. Maximum Dynamic Flow Model. The dynamic network flow model with discrete time setting satisfies Constraints (5)–(8):

$$\sum_{e \in A_v^+, t \geq 0} f_e(t) - \sum_{e \in A_v^-, t - \theta_e \geq 0} f_e(t - \theta_e) = \begin{cases} h_v(t), & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \forall t \in \mathcal{T}, \\ -h_v(t), & \text{if } v \in V^-, \end{cases} \quad (5)$$

$$h_v(t) \geq 0, \quad \forall v \in V, \forall t \in \mathcal{T}, \quad (6)$$

$$u_e(t) \geq f_e(t) \geq 0, \quad \forall e \in A, \forall t \in \mathcal{T}, \quad (7)$$

$$f_e(t) = 0, \quad \forall e \in A, \ t = \overline{T - t_e + 1, T}. \quad (8)$$

Constraint (8) ensures that the flow does not enter arc e at time t if it will have to leave the arc after the given time horizon. The maximum dynamic flow that can enter the arc e within each integral time step t is bounded by the time varying capacity $u_e(t)$; this is ensured by Constraint (7). Flow conservation conditions are ensured in Constraint (5). Flow value at T is defined in (9) and is to be maximized for the MDF:

$$\sum_{t=0}^{T} \sum_{v \in V^+} h_v(t). \quad (9)$$

Multiterminal network for single commodity flow can be reduced to the standard two-terminal network by introducing one virtual source node and one virtual sink node. Virtual arcs connect the new source to true sources and true sinks to the new sink. The transit times of these virtual arcs are zero. The capacities of arcs connecting to the virtual source with all other sources are bounded by the capacities of these sources. The capacities of arcs connecting to virtual sink from true sinks are bounded by the capacities of these sinks. If $\theta_e = 0$ for all $e \in A$ and $T = 0$, then the formulated problem reduces to the classical maximum flow problem on a static network.

2.1.3. Time-Expanded Flow Model. The dynamic network flow problem in $N = (V, A, u_e(t), \theta_e, T)$ can be reduced to a static network flow problem in the time-expanded network $N^T = (V^T, A^T, u^T)$, which is a static representation of the dynamic network. Construction of time-expanded network is as follows:

$$V^T := \{v(t) \colon v \in V, t \in T\},$$
$$A^T := \{(v(t), w(t + \theta_e)) \colon e = (v, w) \in A, 0 \leq t \leq T - \theta_e\}, \quad (10)$$

with capacity function $u_{e(t)}^T := u_e(t)$, for $e(t) \in A^T$. Let $(v(t), w(t + \theta_e)) \in A^T$ and let $f_e(t)$ be a flow in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$. The corresponding flow function in the time-expanded network $N^T = (V^T, A^T, u^T)$ is defined by

$$f_{e(t)}^T = f_{(v(t), w(t+\theta_e))} = f_e(t), \quad \forall e(t) \in A^T. \quad (11)$$

Relation (11) is a bijection from the set of feasible flows in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$ to the set of feasible flows in the time-expanded network $N^T = (V^T, A^T, u^T)$ so that dynamic flows $f_e(t)$ with time horizon T are equivalent to static flows $f_{e(t)}^T$ in the time-expanded network [17].

2.1.4. Minimum Cost Flow Model. Let $d \colon A \longrightarrow Z$ be supply-demand function with supply $d_v > 0$ and demand $d_v < 0$ for $v \in V$, and let $c \colon A \longrightarrow Z$ be the cost function. Then, the minimum cost flow formulated as in [1] is

$$\min \sum_{e \in A} c_e f_e, \quad (12)$$

$$\sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e = d(v), \quad \forall v \in V, \quad (13)$$

$$u_e \geq f_e \geq 0, \quad \forall e \in A. \quad (14)$$

The conservation and capacity constraints are given by (13) and (14), respectively. Equation (15) represents the feasibility of supplies and demands. This constraint implies that the total supply is equal to the total demand:

$$\sum_{v \in V} d_v = 0. \quad (15)$$

The feasibility of minimum cost flow problem can be determined by solving a maximum flow problem [1, 18]. For this, one introduces a supersource node s^*, a supersink node t^*, source arcs (s^*, v) with capacities d_v for $v \in V^+$, and sink arcs (v, t^*) with capacities $-d_v$ for $v \in V^-$. Recall that $\sum_{v \in V^+} d_v = \sum_{v \in V^-} d_v$ holds. If the maximum flow saturates all the source arcs, the minimum cost flow problem is feasible; otherwise, it is infeasible.

3. Maximum Flow Improvement Problems

3.1. Static Flow Improvements. In the maximum static flow improvement problem (MSFIP), an additional nonnegative number U_e, for each $e \in A$, is given so that the capacity u_e

can be increased with some nonnegative cost up to the upper bound $U_e \geq u_e$. The improvement capacity function with nonnegative unit cost b_e is defined as $I: A \longrightarrow Q_{\geq 0}$. The objective of the problem is to maximize the flow from the sources to sinks by increasing the capacities of arcs within the budget restriction where incremental cost is to be accepted to increase arc capacity. Flow improvement problems (16)–(22) have been formulated as in [5]:

$$\sum_{e \in A_v^+} f_e - \sum_{e \in A_v^-} f_e = \begin{cases} h_v, & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \\ -h_v, & \text{if } v \in V^-, \end{cases} \quad (16)$$

$$h_v \geq 0, \quad \forall v \in V, \quad (17)$$

$$I_e \geq 0, \quad \forall e \in A, \quad (18)$$

$$u_e + I_e \geq f_e \geq 0, \quad \forall e \in A, \quad (19)$$

$$U_e \geq u_e + I_e \geq 0, \quad \forall e \in A, \quad (20)$$

$$\sum_{e \in A} I_e b_e \leq B. \quad (21)$$

Constraint (19) controls the arc flows, Constraint (20) limits capacity increment, and Constraint (21) bounds the total capacity incremental cost B with b as the unit cost. By denoting F_{st} to be the sum of flow out from the source that enter into the sink, the objective function equals

$$F_{st} = \sum_{v \in V^+} h_v. \quad (22)$$

Problem 1. The MSFIP with capacity improvement cost is maximum static flow problems (16)–(22), where capacity of arcs with unit costs can be increased up to specified limit bounded by the improvement cost.

The MSFIP with continuous improvement strategy, in which the improvement function $I(e)$ takes any rational values respecting the upper bound, can be solved optimally in polynomial time. The integral MSFIP that takes only integral improvements can be transformed into a budget constraint minimum cost flow problem in polynomial time.

Assume that the unit cost b_e of increasing the capacity of arc e is a nonnegative number and also assume that the optimal flow improvement corresponding to the arc e is $I_e^* = \max\{0, f_e^* - u_e\}$, where f^* is the optimal improved maximum flow. Otherwise, the strategy would waste the cost. Such a behavior can be modeled by a flow cost c_e defined as

$$c_e(f) = \begin{cases} 0, & \text{for } 0 \leq f \leq u_e, \\ b_e \cdot (f - u_e), & \text{for } u_e < f \leq U_e, \end{cases} \quad \forall e \in A. \quad (23)$$

By definition, c_e is a piecewise-linear convex function. In this case, each $e \in A$ is replaced by two parallel arcs e_0 and e_1 to make it linear. The capacities $\overline{u}_e$ and linear costs $\overline{c}_e$ of these arcs are set as

$$\begin{aligned} \overline{u}_{e_0} &:= \overline{u}_e, \\ \overline{u}_{e_1} &:= U_e - u_e, \\ \overline{c}_{e_0} &:= 0, \\ \overline{c}_{e_1} &:= b_e. \end{aligned} \quad (24)$$

This construction is valid from the convexity of cost function. Thus, the improved maximum flow value F_{st}^* can be obtained by solving minimum cost flow with cost at most B by the method of binary search in $[0, nU_{\max}]$.

Theorem 1 (see [5]). *The integral MSFIP can be solved optimally in polynomial time by $O(\log(nU_{\max}))$ minimum cost flow computations in a directed network with $2m$ arcs, where $U_{\max} = \max\{U_e: e \in A\}$ is the maximum capacity.*

Instead of performing a binary search on the interval $[0, nU_{\max}]$, the interval can be searched only in multiplicative steps of $1 + \epsilon$, where $\epsilon > 0$ is a fixed accuracy parameter. The value F'_{st} found by this modified binary search satisfies $F'_{st} \geq (F_{st}^* / (1 + \epsilon))$.

Theorem 2 (see [5]). *Let $\epsilon > 0$ be fixed. Then, a $(1 + \epsilon)$-approximation for MSFIP can be obtained in $O(\log \log_{1+\epsilon}(nU_{\max}))$ minimum cost flow computations in the directed graph with $2m$ arcs.*

Each of the minimum cost flow computations have to be carried out in a graph with $O(gm)$ arcs, where g is the maximum number of breakpoints occurring in the piecewise-linear cost functions. Furthermore, one can solve the problem in strongly polynomial time applying Megiddo's parametric search [19]. The same search can be applied to solve the rational MSFIP, too.

The flow improvement strategy, either to increase the capacity of each arc to its maximum capacity or leave the capacity of arc unchanged, is $\mathcal{NP}$-hard. This 0/1 MSFIP is equivalent to the maximum flow problem with fixed cost on arcs. For given nonnegative capacity u_e and cost c_e on each $e \in A$, the latter problem asks to find a subset A^* of A such that $\sum_{e \in A^*} c_e \leq B$ and the source-sink flow is maximized. But, the decision variant of this problem is $\mathcal{NP}$-complete. We state the following.

Theorem 3 (see [5]). *The maximum flow problem with fixed cost on arcs is $\mathcal{NP}$-hard even on series-parallel and bipartite graphs.*

The pseudo-polynomial time algorithm for the maximum fixed cost flow problem on series-parallel graphs is presented. They are converted into a fully polynomial time approximation scheme by the scaling technique.

Theorem 4 (see [5]). *When the problems are restricted to series-parallel graphs, the maximum flow problem with fixed cost on arcs and 0/1 MSFIP can be solved with fully polynomial time approximation scheme.*

3.2. Dynamic Flow Improvements. This section extends the MSFIP to maximum dynamic flow improvement problem (MDFIP) considering the time factor. Let $I(t)$ and $b(t)$ be capacity improvement and improvement unit cost functions in dynamic network, respectively. The proposed MDFIP can be formulated as follows:

$$\max \sum_{t=0}^{T} \sum_{v \in V^+} h_v(t), \tag{25}$$

$$\text{subject to} \sum_{e \in A_v^+} f_e(t) - \sum_{e \in A_v^-, t-\theta_e \geq 0} f_e(t-\theta_e) = \begin{cases} h_v(t), & \text{if } v \in V^+, \\ 0, & \text{if } v \in V^0, \ \forall t \in \mathcal{T}, \\ -h_v(t), & \text{if } v \in V^-, \end{cases} \tag{26}$$

$$h_v(t) \geq 0, \quad \forall v \in V, \ \forall t \in \mathcal{T}, \tag{27}$$

$$I_e(t) \geq 0, \quad \forall e \in A, \ \forall t \in \mathcal{T}, \tag{28}$$

$$f_e(t) = 0, \quad \forall e \in A, t = \overline{T - t_e + 1, T}, \tag{29}$$

$$u_e(t) + I_e(t) \geq f_e(t) \geq 0, \quad \forall e \in A, \ \forall t \in \mathcal{T}, \tag{30}$$

$$U_e(t) \geq u_e(t) + I_e(t) \geq 0, \quad \forall e \in A, \ \forall t \in \mathcal{T}, \tag{31}$$

$$\sum_{t=0}^{T} \sum_{e \in A} I_e(t) b_e(t) \leq B. \tag{32}$$

This is an extension of MDF where Budget Constraint (32) has been imposed to bound the capacity improvement cost, Constraint (31) deals with the maximum possible improvement, and Constraint (30) deals with the feasibility of flow with improved capacity.

Let $(v(t), w(t+\theta_e)) \in A^T$ and let $I_{e(t)}$ be the capacity improvement on arc $e \in A$; then, the capacity improvement function could be defined as $I_{e(t)}^T = I_{(v(t),w(t+\theta_e))} = I_e(t)$, for all $e(t) \in A^T$. Each dynamic network can be transformed into the corresponding time-expanded network for a given time horizon. The maximum dynamic flow in the given network is equal to the maximum flow in time-expanded network [1].

Problem 2. The MDFIP with capacity improvement cost is the MDF, where capacities of arcs can be increased up to a specified limit accepting the improvement cost.

Theorem 5. *The integral and continuous flow improvement problems in dynamic network can be solved optimally in pseudo-polynomial time.*

Proof. Let us transform the given integral MDFIP in the dynamic network into the MSFIP in the corresponding static network assuming $V^T := \{v(t): v \in V, t \in T\}$, $A^T := \{(v(t), w(t+\theta_e)): e = (v,w) \in A, 0 \leq t \leq T - \theta_e\}$, $u_{e(t)}^T := u_e(t)$, $I_{e(t)}^T := I_{(v(t),w(t+\theta_e))} = I_e(t)$, and $b_{e(t)}^T := b_e(t) \, \forall e(t) \in A^T$. Let $(v(t), w(t+\theta_e)) \in A^T$ and let $f_e(t)$ be a flow in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$; then, the corresponding flow function in $N^T = (V^T, A^T, u^T)$ can be related by

$$f_{e(t)}^T := f_{(v(t),w(t+\theta_e))} = f_e(t) \text{ for all } e(t) \in A^T. \tag{33}$$

Relation (33) is a bijection from the set of feasible flows in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$ to the set of feasible flows in $N^T = (V^T, A^T, u^T)$ so that dynamic flows $f_e(t)$ with time horizon T are equivalent to static flows $f_{e(t)}^T$ in $N^T = (V^T, A^T, u^T)$. Thus, the flow improvement problem in dynamic network $N = (V, A, u_e(t), \theta_e, T)$ with time horizon T, capacity improvement $I_e(t)$, and improvement cost $b_e(t)$ is equivalent to the flow improvement problem in static network $N^T = (V^T, A^T, u^T)$ with capacity improvement $I_{e(t)}^T$ and improvement cost $b_{e(t)}^T$. As the integral maximum flow improvement problem in static network can also be solved optimally (cf. Theorem 1), the integral MDFIP can also be solved optimally.

With similar arguments, the continuous MDFIP can be solved optimally in pseudo-polynomial time. □

The dynamic improvement and dynamic improvement cost have been considered in (25)–(32). While constructing time-expanded network in the proof of Theorem 5, they are transformed into static improvement and static improvement cost by applying the transformations $I_{e(t)}^T := I_{(v(t),w(t+\theta_e))} = I_e(t)$ and $b_{e(t)}^T := b_e(t) \, \forall e(t) \in A^T$, respectively. If one consider improvement and its cost both static in (25)–(32), the time-expanded network copies the arc with the same improved capacity for every arc of the dynamic network. These modifications ensure that each copy of the same arc is improved by the same amount of capacity and the improvement cost of an arc is counted only once for all of its copies. □

4. Flow Improvements with Arc Switching Costs

Contraflow approach increases the flow value by reversing the directions of arcs towards the sinks as a flow towards the sources is neither preferred nor expected in emergencies. This concept without any reversal cost is first incorporated in [20] and analytically studied in [7], where the arc is reversed with its full capacity or left as it is. However, a reversal may require some switching costs. Different contraflow models and solution procedures with switching costs are considered throughout this section.

Figure 1 explains how contraflow works in time invariant network, where there is no arc reversal cost. Given network $N = (V, A, u_e, \theta_e, T)$, the contraflow uses an auxiliary network $\widetilde{N} = (V, \widetilde{A}, \widetilde{u}, \widetilde{\theta}, T)$, where the set of arcs $\widetilde{A}$ contains $\widetilde{e}$ if $e = (v, w) \in A$ or $e^r = (w, v) \in A$. The capacity and symmetric transit time functions are considered as $\widetilde{u}_{\widetilde{e}} := u_e + u_{e^r}$ and $\widetilde{\theta}_{\widetilde{e}} := \begin{cases} \theta_e & \text{for } e \in A \\ \theta_{e^r} & \text{else} \end{cases} \forall \widetilde{e} \in \widetilde{A}$, respectively. Other parameters of the given network remain the same.

4.1. Maximum Static Flow Problem. The maximum static contraflow problem (MSCFP) introduced in [7] maximizes the source-sink flow value, where directions of arcs have been reversed without considering the reversal costs. In this section, we introduce budget constraint MSCFP and propose its solution procedure for a two-terminal network.

The switching cost contraflow (SCCF) finds a feasible flow in $N = (V, A, u_e)$, where the arc directions can be reversed accepting switching costs whenever flow can be improved. The static SCCF problem has a similar structure as the minimum concave-cost network flow problem [21]. These problems ask to find a feasible flow while minimizing the total cost. The total cost is the sum of concave-costs due to the use of arcs by feasible flow. It can be assumed that the concave-cost per arc consists of a fixed cost, whenever this particular arc is used, and a variable cost depends on the amount of flow. Fixing the variable cost to zero leads to a special problem called minimum cost fixed flow (MCFF) problem. The improvement strategy function I is a 0/1 decision if additional capacity is used or not, independent of how much additional capacity is used. An integral flow improvement strategy is considered throughout this section.

Problem 3. The MSCFP with switching cost is the maximum flow problem in the static network where arcs can be reversed accepting some switching costs and total cost is restricted by a given budget. This problem is denoted by MSCFPWSC (Algorithm 1).

Theorem 6. *Algorithm 1 solves Problem 3 optimally in polynomial time for integral reversed capacity.*

Proof. Step 1 constructs an auxiliary network from the given network which is feasible. Step 2 defines the cost function for the improved capacity which is convex and nonlinear, whereas Step 3 is to linearize the cost function defined in Step 2. Thus, both these steps are feasible. Step 4 finds the optimal flow F^*_{st} in $\widetilde{N} = (V, \widetilde{A}, \widetilde{u}_{\widetilde{e}})$ using the binary search method in $[0, n\widetilde{U}_{\max}]$, and Step 5 reverses the direction of arcs according to the direction of the flow obtained in Step 4. So, Step 4 and Step 5 are also feasible. Hence, the algorithm is feasible.

To restrict on budget, this algorithm only reverses the required capacity of the arc so that $I_e = \max\{0, f_{\widetilde{e}} - u_{\overrightarrow{e}}\}$, where $f_{\widetilde{e}}$ represents the flow through the arc $\widetilde{e}$. Cost for the reversed capacity is defined in Step 2 which is not linear but convex and is linearized in Step 3, as in [5, 22]. Then, MSCFP with switching cost is equivalent to the budget constraint maximum flow improvement problem with integral improvement which is equivalent to the budget constraint minimum cost flow problem. Thus, Problem 3 is equivalent to the budget constraint minimum cost flow problem. Suppose F^*_{st} be the optimal integral flow achievable on budget B which can be obtained by the method of binary search in $[0, n\widetilde{U}_{\max}]$ [5].

Step 1 constructs an auxiliary network in linear time and is similar for Step 2 and Step 3. Step 5 reverses the arc directions in linear time according to the flow obtained in Step 4 since not both of the arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be switched at a time. Thus, running time of the algorithm depends on Step 4. Hence, Algorithm 1 solves Problem 3 in $O(\log(n\widetilde{U}_{\max}))$ minimum cost flow computations. □

Algorithm 1 reverses the direction of arcs with integral capacity; thus, the improvement is integral. This implies that the optimal flow of the problem is integral. For this, the binary search algorithm can test only for the integral flow values of the interval. Instead of applying binary search in the interval, one can apply Megiddo's parametric search [19] to extract the solution in Step 4 of Algorithm 1.

The maximum value F'_{st} satisfying the budget constraint lies in $\{1, 1 + \epsilon, \ldots, (1 + \epsilon)^k\}$, where $k = \lceil \log_{1+\epsilon}(n\widetilde{U}_{\max}) \rceil$ and $\epsilon > 0$ is a fixed accuracy parameter. This modified binary search finds new value such that $F'_{st} \geq (F^*_{st}/(1 + \epsilon))$. If such modification is applied in Step 4 of Algorithm 1, a $(1 + \epsilon)$ approximation for budget constrained maximum contraflow can be obtained in $O(\log \log_{1+\epsilon}(n\widetilde{U}_{\max}))$ minimum cost flow computations, where $\epsilon > 0$ is a fixed accuracy parameter.

Example 1. Let $B = 20$ be the total reversal budget for the evacuation network as shown in Figure 2. Before contraflow configuration, 10 units of flow could be reached in the sink from the source through the paths $s - x - t$ and $s - y - t$. Likewise, 19 units of flow could be sent through paths $s - x - t$, $s - y - t$, and $s - y - x - t$ (cf. Figure 2(b)) applying the contraflow approach for which $B = 37$ is required for the complete arc reversals which is more than the given budget. This means that the upper bound of the budget constraint contraflow is 37. The budget constraint contraflow can be found applying Algorithm 1.

The optimal budget constraint flow in the auxiliary network in Figure 1(b) corresponding to Figure 2(a) can be obtained using Step 4. In this network, $\widetilde{U}_{\max} = \max\{\widetilde{u}_{\widetilde{e}} | \widetilde{e} \in \widetilde{A}\} = \max\{10, 9, 3, 10, 9\} = 10$. Thus, the flow can be obtained in $[0, 4 \times 10] = [0, 40]$ as $n := |V|$. Here, the midpoint of the interval is 20 but it is infeasible so that the flow should be

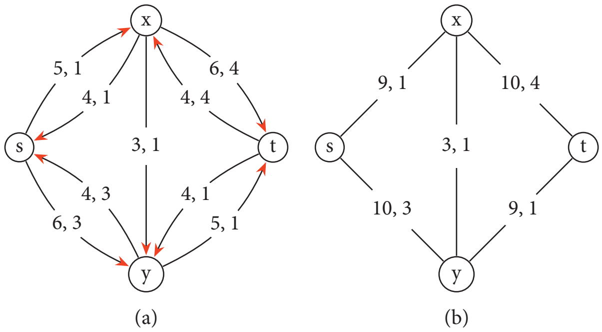


Figure 1: Network with capacities and transit times, respectively. (a) Evacuation network. (b) Auxiliary network.

Input: two-terminal arc reversible network $N = (V, A, u_e)$ with reversal budget B and reversal cost b_e

(1) Construct an auxiliary network, $\widetilde{N} = (V, \widetilde{A}, \widetilde{u}_{\widetilde{e}})$ with new capacity $\widetilde{u}(\widetilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$

(2) Define the cost function by $c_{\widetilde{e}}(f) = \begin{cases} 0 & \text{for } 0 \le f_{\widetilde{e}} \le u_{\overrightarrow{e}} \\ b_{\overleftarrow{e}} \cdot (f_{\widetilde{e}} - u_{\overrightarrow{e}}) & \text{for } u_{\overrightarrow{e}} < f_{\widetilde{e}} \le \widetilde{u}_{\widetilde{e}} \end{cases}$ for all $\widetilde{e} \in \widetilde{A}$

(3) Transform the cost and the capacity functions by

$$\overline{u}_{e_0} := u_{\overrightarrow{e}} \quad \overline{u}_{e_1} := \widetilde{u}_{\widetilde{e}} - u_{\overrightarrow{e}}$$
$$\overline{c}_{e_0} := 0 \quad \overline{c}_{e_1} := b_{\overleftarrow{e}}$$

(4) Find the optimal flow F^*_{st} in the auxiliary network considering costs defined in Step 3 and the total budget B, by applying binary search in $[0, n\widetilde{U}_{\max}]$, where $\widetilde{U}_{\max} = \max\{\widetilde{u}_{\widetilde{e}}: \widetilde{e} \in \widetilde{A}\}$

(5) An arc $\overleftarrow{e} \in A$ is reversed with capacity $f_{\widetilde{e}} - u_{\overrightarrow{e}}$ if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a nonnegative flow along the arc $\overrightarrow{e} \notin A$

Output: maximum contraflow value in $N = (V, A, u_e)$ using arc reversal budget B

Algorithm 1: Maximum static contraflow with switching cost (MSCFAWSC).

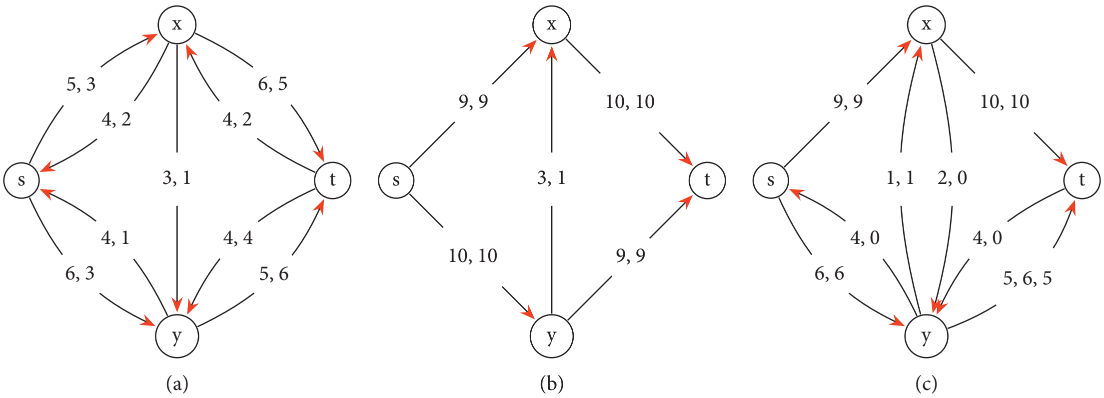


Figure 2: Evacuation network, contraflow without budget constraint and contraflow with budget constraint, respectively. (a) Capacity, unit cost. (b, c) Capacity, flow.

searched in the lower interval $[0, 19]$. Again, a middle value of the interval is 10 and its minimum cost is zero as 5 units of flow can be sent through both paths $s - x - t$ and $s - y - t$ without reversing any arcs. The objective of the problem is to maximize the flow under the given budget so that flow should be searched in the upper interval $[11, 19]$. The middle value is 15. Its minimum cost is 17 as the reversed arcs are (x, s), (t, x), and (x, y) with capacities 4, 4, and 1, respectively. Here, we reverse the arc (x, y) only with capacity 1 as the budget should not be wasted. The resulting network flow is shown in Figure 2(c). Next, the possible upper value should be searched in $[16, 19]$. Let us check the cost for 17. For this, the minimum cost is 27 which exceeds the given budget, and thus, it is infeasible. Similarly, the minimum cost is checked for 16 units of flow which is 22. Again, this exceeds the budget. Hence, the budget constraint contraflow is 15 which is shown in Figure 2(c) for which 17 units of budget have been used. Still, 3 units of budget remain unused but they are insufficient.

Theorem 7. *The fixed switching cost contraflow problem is equivalent to* 0/1-*MSFIP where all data to be positive integral* [7].

Input: two-terminal arc reversible network $N = (V, A, u_e(t), \theta_e, T)$ with reversal budget B, reversal cost $b_e(t)$, and symmetric arc capacity

(1) Transform the given dynamic network into the time-expanded network by
$V^T := \{v(t): v \in V, t \in T\}$
$A^T := \{(v(t), w(t+\theta_e)) \mid e = (v,w) \in A, 0 \le t \le T - \theta_e\}$
$u^T_{e(t)} := u_e(t)$, for $e(t) \in A^T$
$b^T_{e(t)} := b_e(t)$, for $e(t) \in A^T$
capacities and costs of holdover arcs are infinity and zero, respectively

(2) Construct an auxiliary $\widetilde{N}^T = (V^T, \widetilde{A}^T, \widetilde{u}^T)$ as $\widetilde{e}(t) \in \widetilde{A}$ if $\overrightarrow{e}(t) = (v(t), w(t+\theta_e)) \in A^T$ or $\overleftarrow{e}(t) = (w(t-\theta_e), v(t)) \in A^T$ with new capacity $\widetilde{u}^T_{\widetilde{e}(t)} = u^T_{\overrightarrow{e}(t)} + u^T_{\overleftarrow{e}(t)}$ for all $\widetilde{e}(t) \in \widetilde{A}^T$

(3) Define the cost function by $c^T_{\widetilde{e}(t)}(f^T_{\widetilde{e}(t)}) = \begin{cases} 0 & \text{for } 0 \le f^T_{\widetilde{e}(t)} \le u^T_{\overrightarrow{e}(t)} \\ b^T_{\overrightarrow{e}(t)} \cdot (f^T_{\widetilde{e}(t)} - u^T_{\overrightarrow{e}(t)}) & \text{for } u^T_{\overrightarrow{e}(t)} < f^T_{\widetilde{e}(t)} \le \widetilde{u}^T_{\widetilde{e}(t)} \end{cases}$ for all $\widetilde{e}(t) \in \widetilde{A}^T$

(4) Transform the cost and the capacity functions by $\begin{matrix} \overline{u}^T_{e_0} := u^T_{\overrightarrow{e}(t)} & \overline{u}^T_{e_1} := \widetilde{u}^T_{\widetilde{e}(t)} - u^T_{\overrightarrow{e}(t)} \\ \overline{c}^T_{e_0} := 0 & \overline{c}^T_{e_1} := b^T_{\overleftarrow{e}(t)} \end{matrix}$

(5) Find the optimal flow in the auxiliary network considering budget B and the cost defined in Step 4 using binary search in $[0, nT\widetilde{U}_{\max}]$, where $\widetilde{U}_{\max} = \max\left\{\widetilde{u}^T_{\widetilde{e}(t)} : \widetilde{e}(t) \in \widetilde{A}^T\right\}$

(6) An arc $\overleftarrow{e}(t) \in A^T$ is reversed with capacity $\widetilde{f}^T_{\widetilde{e}_{(t)}} - u^T_{\overrightarrow{e}(t)}$ if and only if the flow along $\overrightarrow{e}(t) \in A^T$ is greater than $u^T_{\overrightarrow{e}(t)}$ or there is a nonnegative flow along the arc $\overrightarrow{e}(t) \notin A^T$

(7) Reconstruct the dynamic network flow from the static network flow according to the flow obtained in Step 5 and arc reversal in Step 6

Output: Maximum dynamic contraflow in $N = (V, A, u_e(t), \theta_e, T)$ with respect to budget B

ALGORITHM 2: Maximum dynamic contraflow with switching cost (MDCFAWSC).

Since the 0/1 maximum flow improvement is equivalent to the maximum fixed cost flow problem, this implies that the FSCCF is $\mathcal{NP}$-hard to solve (Theorem 3). The fixed cost for arc reversals makes the problem $\mathcal{NP}$-hard, even in the static case. The static contraflow algorithm is 'blind' for the arc reversal decisions. Adding a time component to FSCCF makes it practically even more difficult to solve. Based on the above results, the following theorems are proposed.

Theorem 8. *The FSCCF problem is $\mathcal{NP}$-hard even in series-parallel graphs.*

Proof. The FSCCF problem is equivalent to 0/1-MSFIP (Theorem 7). As the maximum flow problem with fixed cost on arcs is equivalent to 0/1-maximum flow improvement [5], then by Theorem 3, it can be claimed that the FSCCF problem is $\mathcal{NP}$-hard even in series-parallel graphs. ☐

Theorem 9. *There is a fully polynomial time approximation scheme for the FSCCF problems when the problems are restricted to series-parallel graphs.*

Proof. As in proof of Theorem 8, the FSCCF problem is equivalent to 0/1-MSFIP and the maximum flow problem with fixed costs on arcs. Since all hardness and approximation results of these problems will be carried over to the FSCCF problem so that a fully polynomial time approximation scheme can be obtained for the FSCCF problems in series-parallel graphs (Theorem 4). ☐

As the fixed switching cost contraflow problem is equivalent to the 0/1 maximum flow improvement which is equivalent to the minimum cost fixed flow problem, an approximation solution can be found by using the cost-to-time ratio approach [5, 18]. This can be obtained by assuming the capacities u_e and unit cost c_e/u_e.

4.2. Maximum Dynamic Flow Problem. The notion of MDF is introduced and solved in [7] by reversing the arcs at time zero with zero cost. In this section, reversal cost is considered to maximize the dynamic contraflow subject to budget constraint which is an extension of MSCFP from static to dynamic networks.

Problem 4. The MDCF problem with switching cost is the MDF where arcs can be reversed accepting switching cost such that the total cost is subject to the budget constraint. It is denoted by MDCFPWSC.

Theorem 10. *Algorithm 2 solves Problem 4 optimally in pseudo-polynomial time.*

Proof. Step 1 is well defined as it transforms the given dynamic network flow problem into the static network flow problem. The feasibility of other steps of the algorithm can be shown as in Theorem 6. Thus, the algorithm is feasible.

The dynamic network can be transformed into static network as in Step 1. Step 2 constructs the auxiliary network, and Step 6 finds an optimal flow in auxiliary network with respect to the cost defined in Step 5 and total budget B. Let $f_e(t)$ be the budget constraint contraflow in the dynamic network. Then, the corresponding budget constraint flow function in the network $\widetilde{N}^T = (V^T, \widetilde{A}^T, \widetilde{u}^T)$ can be related by the following equation:

$$f^T_{\tilde{e}(t)} = f_{\left(v(t),w\left(t+\theta_{\tilde{e}}\right)\right)} = f_e(t), \quad \forall \tilde{e}(t) \in \tilde{A}^T. \tag{34}$$

It is a bijection from the set of feasible budget constraint contraflows in the dynamic network $N = (V, A, u_e(t), \theta_e, T)$ to the set of feasible budget constraint flows in $\tilde{N}^T = (V^T, \tilde{A}^T, \tilde{u}^T)$ so that the contraflows $f_e(t)$ with time horizon T are equivalent to the flows $f^T_{\tilde{e}(t)}$ in the time-expanded network. Remaining proof of the theorem can be shown as in Theorem 6. Suppose F^*_{dy} be the optimal contraflow achievable on budget B. Hence, the optimal contraflow can be obtained in $O(\log(Tn\tilde{U}_{\max}))$ minimum cost flow computations. □

5. Conclusions

To find the minimum cost, a maximum flow and the quickest time in static or dynamic networks are still emerging fields of research in the literature. Moreover, the best reconfiguration of an evacuation network is being more interesting both from theoretical as well as application points of view. However, in most of the cited publications, the arc reversal costs are considered to be zero, given the highest priority for evacuees rather than taking account of a fixed budget in emergency periods. Many efficient algorithms are presented so far.

In this paper, we look at the above problems considering the fixed budgetary costs in addition. Our focus is given on the flow improvement problems with switching costs constrained by arc reversal costs on different static and dynamic network flow models. Different variants of flow improvement problems have been studied, and their complexities are given. Here, the contraflow reconfiguration with arc switching cost has been introduced for the first time. Based on this approach, a polynomial time algorithm and a pseudo-polynomial time algorithm are proposed for maximum static and maximum dynamic contraflow improvement problems, respectively. The arc capacities in these dynamic flow models are subject to the time-dependent capacity constraints with time invariant transit times. Therefore, these solutions are equally applicable if both these parameters are taken to be time invariant. If both parameters are taken to be time- or flow-dependent, the hardness increases, and it is our further interest. The findings of these investigations are of both theoretical and practical interest.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

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

Acknowledgments

The first author thanks the University Grants Commission, Nepal, for the PhD research fellowship. The authors would also like to thank the anonymous referees and the editor for their valuable suggestions to improve the quality of this paper.

References

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

[2] L. Fleischer and E. Tardos, “Efficient continuous-time dynamic network flow algorithms,” *Operations Research Letters*, vol. 23, no. 3–5, pp. 71–80, 1998.

[3] H. W. Hamacher and S. A. Tjandra, “Mathematical modeling of evacuation problems: a state of the art,” in *Pedestrian and Evacuation Dynamics*, M. Schreckenberger and S. D. Sharma, Eds., Springer, Berlin, Germany, 2001.

[4] 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.

[5] S. O. Krumke, H. Noltemeier, S. Schwarz, H.-C. Wirth, and R. Ravi, “Flow improvement and network flows with fixed costs,” in *Proceedings of the International Conference of Operations Research (OR’98)*, pp. 158–167, Zürich, Switzerland, August 1999.

[6] S. Kim, S. Shekhar, and M. Min, “Contraflow transportation network reconfiguration for evacuation route planning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 1115–1129, 2008.

[7] 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.

[8] U. Pyakurel and T. N. Dhamala, “Continuous time dynamic contraflow models and algorithms,” *Advances in Operations Research*, vol. 2016, Article ID 7902460, 7 pages, 2016.

[9] U. Pyakurel and T. N. Dhamala, “Evacuation planning by earliest arrival contraflow,” *Journal of Industrial & Management Optimization*, vol. 13, no. 1, pp. 489–503, 2017.

[10] U. Pyakurel, H. N. Nath, and T. N. Dhamala, “Efficient contraflow algorithms for quickest evacuation planning,” *Science China Mathematics*, vol. 61, no. 11, pp. 2079–2100, 2018.

[11] U. Pyakurel, H. N. Nath, S. Dempe, and T. N. Dhamala, “Efficient dynamic flow algorithms for evacuation planning problems with partial lane reversal,” *Mathematics*, vol. 7, no. 10, p. 993, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] J. Hua, G. Ren, Y. Cheng, and B. Ran, “An integrated contraflow strategy for multimodal evacuation,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 159473, 10 pages, 2014.

[16] J. Mo, M. Gao, and L. Liu, “An improved critical-edge model for finding optimal contraflow links considering the influence of intersections,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 8681417, 7 pages, 2019.

[17] M. A. Fonoberova and D. D. Lozovanu, “The maximum flow in dynamic networks,” *Computer Science Journal of Moldova*, vol. 12, no. 3, p. 36, 2004.

[18] R. K. Ahuja, T. L. Mangati, and J. B. Orlin, *Networks Flows: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
[19] N. Megiddo, "Combinatorial optimization with rational objective functions," *Mathematics of Operations Research*, vol. 4, no. 4, pp. 414–424, 1979.
[20] S. Kim and S. Shekhar, "Contraflow network reconfigurations for evacuation planning: a summary of results," in *Proceeding of the 13th ACM Symposition on Advances in Geographic Information System, GIS '05*, vol. 13, pp. 250–259, Bremen, Germany, November 2005.
[21] G. M. Guisewite and P. M. Pardalos, "Minimum concave-cost network flow problems: applications, complexity, and algorithms," *Annals of Operations Research*, vol. 25, no. 1, pp. 75–99, 1990.
[22] S. Schwarz and S. O. Krumke, "On budget-constrained flow improvement," *Information Processing Letters*, vol. 66, no. 6, pp. 291–297, 1998.

International Journal of Operations Research Vol. 16, No. 1, 13-26 (2019)

Maximum FlowLoc Problems with Network Reconfiguration

Ram Chandra Dhungana and Tanka Nath Dhamala*

Central Department of Mathematics, Tribhuvan University, P.O. Box. 143143, Kathmandu, Nepal

Received February 2019; Revised March 2019; Accepted March 2019

Abstract: The large scale calamities caused by natural or human-created disasters are the challenging issues to protect the life and their surroundings. It is a complex task to develop the significant and universally accepted solution strategy for handling these issues. Appropriate facility locations and transportation facilities are essential components for the efficient evacuation planning flow models. Contraflow, the lane reversal strategy, is one of the widely accepted solution approach for evacuation planning as it maximizes the outbound capacities of roads by reversing the required road directions and makes the traffic smooth. This significantly increases the flow value and decreases the evacuation time. In this work, the network facility location and the contraflow approach are incorporated to the flow models and some efficient algorithms are presented to locate the facility with an objective of minimum flow loss on the evacuation network. Our facility location contraflow solutions obtain the optimal plans with respect to the given and arbitrary locations.
Keyword — Evacuation plan, dynamic network, flow model, facility location, contraflow, ContraFlowLoc

1. INTRODUCTION

There has been a growing number of natural as well as man-made disasters worldwide. An essential aspect of the awareness to such events is the choice of shelter locations such that everyone who has to be evacuated can be accommodated. Nevertheless, one must take into consideration the traffic that will result from evacuation network as it has a big influence on the duration of the evacuation process. Due to the importance of the best possible choice of shelter locations with minimum flow loss or minimum increase in the network clearance time, the efficient evacuation planning has drawn the interest for current researchers.

An evacuation network represents the intersections of roads (i.e., rooms in a building or intersection of streets in a region) as nodes and road segments between nodes (i.e., doors between rooms, or streets in region) as arcs. The sources represent initial places where evacuees are located and start to move and the sinks represent safe places where they are supposed to arrive. Most traffic delays occur in roads due to different facilitity locations around the roads. Shifting people with efficient routes to the sinks with proper facility locations is an innermost challenge to manage a regional evacuation plan. In order to cope such real-world problems, the decision maker has to target more than one objectives or consider different factors or measures, then the problems should be changed into multi-criteria decision making problems.

After the development of maximal static and maximum dynamic flow models and the respective algorithms by Ford and Fulkerson (1958), various network flow problems are studied that are applicable for evacuation planning. For example, the maximum dynamic flow problem to shift maximal amount in a given time, the earliest arrival flow problem to maximize the number of evacuees in every possible time, the quickest flow problem for allocating the evacuees to a safer zone in minimum time and the lexicographically maximum dynamic flow problem to send maximum number of evacuees in given priority within the given time period. We refer to Dhamala, Pyakurel, and Dempe (2018) for the summery of results and the extensive references. Philpott (1982) firstly introduced network flow model for continuous time setting. The efficient continuous-time dynamic network flow algorithms have been developed by Fleischer and Tardos (1998) using the natural transformation.

The location theory was introduced by Weber (1909) with applications for industries. Different discrete network location models and algorithms have been investigated by Daskin (1995). For details, we refer to Drezner and Hamacher (2002). The influence of facilities on the walking speed, the walking behaviour of pedestrians, the necessity of placing security personnels to guide the pedestrians to the locations, different positions of facilities and their influence on the

*Corresponding author's e-mail: amb.dhamala@daadindia.org

behaviour and walking speed of pedestrians have been investigated. However, Hamacher, Heller, and Rupp (2013) uses location theory to improve the existing evacuation models where two different models, network flow and location theory have been integrated to introduce FlowLoc theory in evacuation modeling.

Contraflow approach is another emerging and widely accepted model for evacuation planning that increases the outbound road capacities by reversing the direction of roads towards the safe destinations. For the large scale evacuations, the evacuation time has been improved by at least 40 percent with at most 30 percent of the total arc reversals, Kim, Shekhar, and Min (2008). Different contraflow models with their efficient solution algorithms and a greedy heuristic have been developed after the first integer programming formulation (Kim et al., 2008). They showed that the general problem of minimizing the evacuation time is $\mathcal{NP}$-hard. Pyakurel and Dhamala (2016) introduced the first contraflow model for continuous time setting. They have presented efficient algorithms to solve the maximum continuous dynamic contraflow, quickest continuous contraflow and earliest arrival contraflow problems.

A polynomial time algorithm to solve the abstract maximum dynamic contraflow with path reversal capability in continuous time setting has been presented by Pyakurel, Dhamala, and Dempe (2017). Dhungana, Pyakurel, and Dhamala (2018) introduced the abstract contraflow problems and their solution procedures for discrete time setting with path reversals. The quickest contraflow problems with constant and load dependent transit times, their algorithms and computational experiments have been discussed by Pyakurel, Nath, and Dhamala (2018a). Various network flow models and solutions with partial contraflow based on path reversals have been presented by Pyakurel, Nath, and Dhamala (2018b). For the important property of the contraflow reconfiguration and the analytical solutions for various contraflow problem, we refer to Dhamala et al. (2018) and the references therein. The contraflow problem can be solved with the same complexity as without contraflow but flow value may be up to double and the evacuation time the half.

Xie and Turnquist (2011) have introduced and solved the lane based contraflow and crossing elimination strategies at intersections simultaneously. The evacuation network model with lane-based reversal and flow routing has been considered by Zhao, Feng, Li, and Bernard (2016). On the evacuation planning with contraflow and crossing elimination jointly, a bi-level lane-based network optimization and simulation model have been formulated by Xie, Lin, and Waller (2010). The multi-model integrated contraflow for uncertain arrivals of evacuees in an evacuation region with low mobility population is presented by Hua, Ren, Cheng, and Ran (2014). Transit-based models are initiated with vehicle routine problem whereas the integrated strategy contains non-contraflow to shorten the strategy setup time, full-lane contraflow to minimize the evacuation network capacity and bus contraflow to realize the transit cycle operation. The transit priority during multi-modal evacuation can be provided by network aggregation method and an integrated contraflow strategy (Hua et al., 2014).

The plan of the paper is given as follows. The notations and prior works in network flow, FlowLoc and contraflow models are given in Section 2. The set of locations can be fixed in advance or it can be an arbitrary. The former approach has its advantage in allocating facilities at predefined locations that a policy maker requires, however, the latter one gives us a flexibility of allocating a facility among the arbitrary locations. Based on given possible locations, we introduce the two-terminal FlowLoc problem in continuous time dynamic network and propose an efficient algorithm for its solution in Section 3. The corresponding results for two-terminal FlowLoc static and dynamic contraflow problems are presented in Section 4. The results with arbitrary locations are extended in Section 5. Section 6 concludes the paper.

2. DENOTATIONS AND PRIOR WORKS

An evacuation network $N = (V, A, u_e, \theta_e, S, D)$ is a directed graph where V, $A \subseteq V \times V$, S, D, T represent the set of vertices, arcs, sources, sinks and the time horizon, respectively. The capacity and travel time vectors of $e \in A$ are denoted by u_e and θ_e, respectively. The evacuation time is represented by $T_d = \{0, 1, \cdots, T\}$ in discrete model, whereas $T_c = \{[0, 1), \cdots, [T, T+1)\}$ in continuous model. Let $f^c_{\text{dyna}} : A \times T_c \longrightarrow R^+$ and $f^d_{\text{dyna}} : A \times T_d \longrightarrow R^+$ be the dynamic flow function for continuous and discrete time setting, respectively. The incoming and outgoing arcs of the vertex v are denoted and defined by $A^-_v = \{(k, v) \in A \mid v, k \in V\}$ and $A^+_v = \{(v, w) \in A \mid v, w \in V\}$, respectively. It is assumed that the flow is possible only at positive time. The first mathematical formulation for continuous time network flow with $S = \{s\}$ and $D = \{t\}$ is considered by Philpott (1982) which satisfies the following constraints.

$$
\begin{aligned}
\int_0^T \sum_{e \in A_v^-} f_{\text{dyna}}^c(e, \tau - \theta_e) d\tau - \int_0^T \sum_{e \in A_v^+} f_{\text{dyna}}^c(e, \tau) d\tau &= 0, \ \forall v \notin \{s, t\} && (1) \\
\int_0^\tau \sum_{e \in A_v^-} f_{\text{dyna}}^c(e, \tau - \theta_e) d\tau - \int_0^\tau \sum_{e \in A_v^+} f_{\text{dyna}}^c(e, \tau) d\tau &\geq 0, \ \forall v \notin \{s, t\}, \ \tau \in [0, T] && (2) \\
\int_0^T \sum_{e \in A_s^-} f_{\text{dyna}}^c(e, \tau - \theta_e) d\tau - \int_0^T \sum_{e \in A_t^+} f_{\text{dyna}}^c(e, \tau) d\tau &= 0 && (3) \\
0 \leq f_{\text{dyna}}^c(e, \tau) \leq u(e, \tau) \qquad & \forall e \in A, \ \tau \in [0, T]. && (4)
\end{aligned}
$$

The objective of the maximum dynamic flow problem is to maximize the value of flow for given time T satisfying the Constraints (1-4). The maximum flow value at given time T is defined by

$$\text{val}(f_{\text{dyna}}^c, T) = \int_0^T \sum_{e \in A_s^-} f_{\text{dyna}}^c(e, \tau - \theta_e) d\tau = \int_0^T \sum_{e \in A_t^+} f_{\text{dyna}}^c(e, \tau) d\tau \tag{5}$$

If $f : A \longrightarrow R^+$ be the static flow and if $f_{\text{dyna}}^d(\tau)$ be the dynamic flow that enters arc e at time $\tau = 0, 1, \cdots, T-1$, then the following relation holds true.

$$f_{dyna}^d(\tau) = \sum_{\sigma}^{\theta_e - 1} f(\tau - \sigma), \text{ for all}, \tau = 0, 1, \cdots, T-1 \tag{6}$$

The flow that enters e at $\tau - \theta_e$ arrives at the head node at time τ in discrete time, but at time $[\tau + 1)$ in continuous time. Then the continuous dynamic flow f_{dyna}^c is feasible and the amount of source-sink flow at any integer time interval $[\tau, \tau + \gamma)$, for $\tau = 0, 1, \cdots, T-1$, $\gamma \in N$ will be the same for both settings. The dynamic flow models for discrete and continuous time setting have been connected by natural transformation (7) introduced by Fleischer and Tardos (1998).

$$f_{dyna}^c(\alpha) = f_{dyna}^d(\beta) \text{ for all } \beta \text{ and } \alpha \text{ with } \beta \leq \alpha < \beta + 1 \tag{7}$$

Efficient continuous time dynamic network flow algorithms can be found by Fleischer and Tardos (1998), whereas the same solution can be obtained by solving the discrete problem, Equation (7).

The static network flow model can be obtained from the above dynamic flow model by omitting the time factor. The maximum static flow model maximize the value val(f) satisfying Constraints (8 - 9).

$$\sum_{e \in A_v^+} f(e) - \sum_{e \in A_v^-} f(e) = \begin{cases} \text{val}(f) & \text{if } v = s \\ 0 & \text{if } v \neq s, t \\ -\text{val}(f) & \text{if } v = t \end{cases} \tag{8}$$

$$0 \leq f(e) \leq u(e), \quad \forall \ e \in A \tag{9}$$

2.1 FlowLoc

Let $L \subseteq A$ be the set of all feasible locations, P the set of all facilities, $r : P \to N$ the size of the facilities and nol $: L \to N$ the number of facilities that can be placed on the possible locations. The FlowLoc problem asks for an allocation loc $: P \to L$ of the facilities to the arcs, such that the $s - t$ flow value in the network $N_{\text{dyna}}^{\text{loc}} = (V, A, u'_e, \theta_e, s, t)$ is maximized where $u'_e = u_e - \max\{r_p : \text{loc}(p) = e\}$. If more than one facility are placed on location l only the size of the largest facility determines the reduction of the capacity on the arc. Other modelling alternatives for placing more than one facility on an edge has been discussed by Hamacher et al. (2013). The multi-facility FlowLoc problems (q-FlowLoc) find locations for the q facilities $p \in P$ with size r_p such that the reduction of the maximum flow value is as small as possible and not more than nol(l) facilities are placed on each arc $l \in L$. In particular, the Single-FlowLoc problem assigns one facility for $q = 1$ among the given set of facilities.

2.2 Contraflow

In contraflow approach, the auxiliary network of given network will be constructed by adding the capacities of two way arcs and allowing the directions in both ways with symmetric capacities and transit times. The auxiliary network $\tilde{N} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \theta_{\tilde{e}}, S, D)$ is constructed from given evacuation network N as $\overrightarrow{e} = (i,j) \in \tilde{A}$, if $\overrightarrow{e} \in A$ or $\overleftarrow{e} = (j,i) \in A$. The arc capacity function $\tilde{u}$ is given by $\tilde{u}_{\tilde{e}} = u(\overrightarrow{e}) + u(\overleftarrow{e})$ for all arcs $\overrightarrow{e} \in \tilde{A}$. The transit time is defined as follows

$$\tilde{\theta}(\tilde{e}) = \begin{cases} \theta(\overrightarrow{e}) & \text{for } \overrightarrow{e} \in A, \\ \theta(\overleftarrow{e}) & \text{else,} \end{cases} \qquad \forall\, \tilde{e} \in \tilde{A}.$$

3. FLOWLOC PROBLEMS

The location of emergency units or other supports are most affecting factors in the evacuation network. Placing any facilities on arcs can affect the size of maximum flow value and quickest evacuation time. The multi terminal q-FlowLoc problem (q-MT-FlowLoc) is $\mathcal{NP}$-complete, Heller and Hamacher (2011). They provide some heuristic solutions for it. Here, we consider the Single-FlowLoc problem with $q = 1$ and present polynomial time algorithms to solve it.

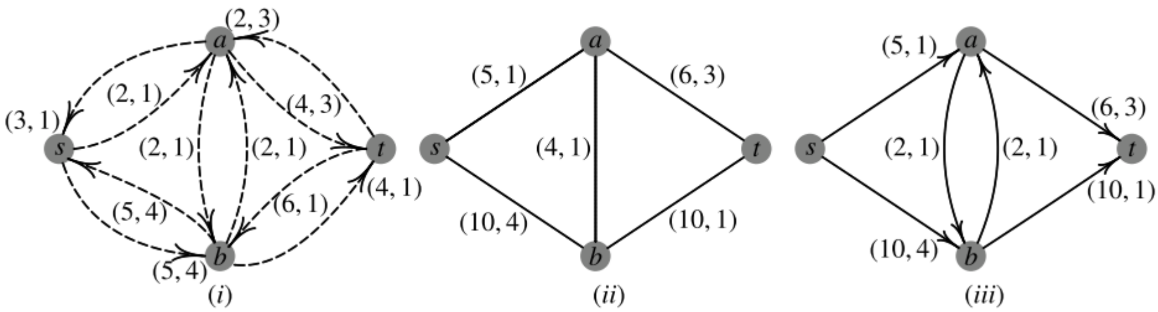


Figure 1: Evacuation, auxiliary and contraflow reconfiguration networks, respectively.

3.1 Maximum Static FlowLoc

The maximum flow can be solved by using labelling algorithm in pseudo-polynomial time $O(mnU)$, where m, n and U represent number of arcs, number of vertices and maximum capacity of arcs, respectively. The shortest augmenting path algorithm finds the shortest augmenting path in polynomial time $O(m^2n)$ from the residual network considering the number of arcs as cost. Several algorithms can be found to improve the complexity of the maximum flow algorithm. We consider the maximum flow algorithm of time complexity $O(nm)$, Orlin (2013). The maximum FlowLoc problem on static network and its solution procedures have been presented by Hamacher et al. (2013). They solved the problem in time $O(|L|n^3)$. But the same problem can be solved using Algorithm 1 in time $O(|L|nm)$. The maximum FlowLoc problem and its efficient solution for the two-terminal static network will be studied in this section.

Problem 1. The maximum FlowLoc problem on static network $N_{stat} = (V, A, u_e, s, t)$ asks to locate the facility in possible locations of the network such that the resulting static flow is maximum in the updated network $N_{stat}^{\text{loc}} = (V, A, u'_e, s, t)$.

Theorem 1. The maximum FlowLoc problem can be solved optimally in time $O(|L|nm)$.

Proof. Algorithm 1 iterates through all possible locations $l \in L$, determines the maximum flow from source to sink if location l hosts facility p and thus finds the optimal location for facility p by comparing all those maximum flow values. For every possible location, maximum flow has to be determined. Thus, Algorithm 1 has the complexity $O(|L|nm)$, where $O(nm)$ is the complexity of a maximum flow algorithm. Hence, Algorithm 1 solves Problem 1 optimally in polynomial time. □

Example 1. Consider the static network given by Figure 1(i) ignoring all arc transit times. Suppose $L = \{(s,a), (s,b), (a,b), (a,t), (b,t)\}$, nol $= 1$ and $r_p = 2$. Before locating facility on the network, 7 units of flow can be sent to the sink through the paths $s - a - t$, $s - b - a - t$ and $s - b - t$ with path flows 2, 2 and 3, respectively. If we pick location (a,b) for the facility it will not reduce any flow from the given network. Thus, optimal flow after providing location remains the same. But it is not necessary that the optimal flow should remain the same in all networks after providing a facility on given network, for example the placement of facility on any other arcs of L reduces the flow value. Thus, the location (a, b) and flow value 7 are the optimal FlowLoc solutions.

Algorithm 1: Maximum Static FloLoc

Input : A directed static network $N_{stat} = (V, A, u_e, s, t)$, locations L, size r_p of facility p
Output: Maximum flow value max_flow, location loc(p) of facility p in the network $N_{stat} = (V, A, u_e, s, t)$

Set: max_flow := -1
  for all $l \in L$ **do**
    if $u_l \geq r_p$ **then**
      $u_l = u_l - r_p$
      max_flow_temp = v(max_flow(N_{stat}))
      $u_l = u_l + r_p$
      if max_flow <max_ flow_temp **then**
        max_flow = max_flow_temp
        loc$(p) = l$
      end if
    end if
  end for
  return max_flow, loc(p)

3.2 Maximum Dynamic FlowLoc

In this section, we introduce FlowLoc problem on dynamic network for continuous time setting that locate given facility in given possible location and maximize the flow on updated dynamic network. We also proposed an efficient algorithm for the problem. Either temporally repeated solution, minimum cost circulation solution considering time as cost or static solution with time expanded network can be used to find the maximum dynamic solution. The complexity of static solution with time expanded network depends in given time horizon so that it is not polynomial. Thus, we have considered the most efficient temporally repeated solution approach which solves the maximum dynamic flow problem in time $O((m \log n)(m + n \log n))$.

Problem 2. The maximum dynamic FlowLoc problem is to locate the facility in a possible location such that the dynamic flow is maximum in $N^{\text{loc}}_{\text{dyna}} = (V, A, u'_e, \theta_e, T, s, t)$.

Algorithm 2: Maximum Dynamic FlowLoc

Input : A dynamic network $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$, locations L, size r_p of the facility p
Output: Maximum flow value max_dyna_flow, location loc(p) of facility p in $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$

Set: max_dyna_flow := -1
for $l \in L$ **do**
  if $u_l \geq r_p$ **then**
    $u_l = u_l - r_p$
    max_dyna_ flow_ temp = v(max_dyna_flow(N_{dyna}))
    $u_l = u_l + r_p$
    if *max_dyna_flow* < *max_dyna_flow_ temp* **then**
      max_dyna_flow = max_dyna_flow_temp
      loc$(p) = l$
    end
  end
end
return max_dyna_flow, loc(p)

Theorem 2. The maximum dynamic FlowLoc problem can be solved optimally.

Proof. Algorithm 2 iterates through all possible locations $l \in L$, gives the maximum dynamic flow value in updated $N^{\text{loc}}_{\text{dyna}} = (V, A, u'_e, \theta_e, T, s, t)$ if location l hosts facility p in given network. The optimal location for facility p is obtained by comparing all those maximum dynamic flow values. Hence, Algorithm 2 solves Problem 2 optimally. □

Corollary 1. Algorithm 2 solves the maximum dynamic FlowLoc problem in polynomial time.

Proof. The complexity of the Algorithm 2 is $O(|L|(m \log n)(m + n \log n))$, where $O((m \log n)(m + n \log n))$ represents the complexity of maximum dynamic flow. □

Example 2. Consider the network of Figure 1(i) with $L = \{(s, a), (s, b), (a, b), (a, t), (b, t)\}$, nol $= 1$, $r_p = 2$ and $T = 8$. The solution can be obtained by applying Algorithm 2. The optimal flow is 27 with optimal location (a, b), shown in Table 1.

Table 1: Maximum dynamic flow corresponding to location (a, b).

Time Horizon	Paths	Length	Flow	Total Dynamic Flow
4	$s - a - t$	4	2	2
5	$s - a - t$	4	2	
	$s - b - t$	5	4	8
6	$s - a - t$	4	2	
	$s - b - t$	5	4	14
7	$s - a - t$	4	2	
	$s - b - t$	5	4	20
8	$s - a - t$	4	2	
	$s - b - t$	5	4	
	$s - b - a - t$	8	1	27

The maximum dynamic flow problem maximizes the solution at given time horizon, where as the earliest arrival flow problem maximizes the solution at every possible time from the beginning. This implies that, every earliest arrival solution is maximum dynamic solution but the converse may not be true. The earliest arrival flow problem has been solved by using the successive shortest augmenting path algorithm (Wilkinson, 1971). The standard chain decomposition could not be adopted to solve the earliest arrival problem as we can not repeat the same path for each time. The non-standard chain decomposition uses the minimum cost maximum flow approach assuming backward chain flows. We can not fix the direction of any arc to get the earliest arrival flow. Thus, the FlowLoc problem could be extended to different dynamic flow problems in various networks, but it can not be extended to the earliest arrival flow models. For the justification we provide following example.

Example 3. Consider the network of Figure 1(iii) with $L = \{(s, a), (s, b), (a, b), (a, t), (b, t)\}$, nol $= 1$ and $r_p = 2$. The earliest arrival flow without any facilities on the network is 65, as shown in Table 2. Applying successive shortest path algorithm on the network, different solutions with respect to different locations have been shown in Table 3 and Table 4. From the table, the facility on the location (a, b) reduces the minimum flow but it is not earliest arrival flow, since 2 units of flow can be sent in $[0, 3]$. If we look for earliest arrival flow we loss the maximality in some interval of time. Again, facility on locations $\{(s, a), (s, b), (a, t), (b, t)\}$ will send 2 units of flow in $[0, 3]$ but could not send as much flow as we send by considering the location (a, b) from some points of time. For example, if we consider location (a, t) we can send 33 units of flow in $[0, 6]$ but by considering location (a, b), 35 units of flow can be sent in the same time.

4. CONTRAFLOWLOC PROBLEMS

In this section, we define both static and dynamic maximum ContraFlowLoc problems and present efficient algorithms to solve them. The proposed maximum ContraFlowLoc algorithms solve the problems in the same time complexity as the maximum FlowLoc problems but the value of flow can be double after contraflow reconfiguration.

4.1 Maximum Static ContraFlowLoc

The maximum static FlowLoc problem locates the facility on possible locations and maximize the flow on updated network whereas the contraflow allows arc reversals to improve the solution. The maximum FlowLoc problem is introduced by Hamacher et al. (2013) and the maximum contraflow problem by Rebennack, Arulselvan, Elefteriadou,

Table 2: Earliest arrival flow on network given in Figure 1 (iii).

Time Horizon	Paths	Length	Flow	Earliest Arrival Flow
3	$s-a-b-t$	3	2	2
4	$s-a-b-t$	3	2	
	$s-a-t$	4	3	7
5	$s-a-t$	4	3	
	$s-b-t$	5	10	20
6	$s-a-t$	4	5	
	$s-b-t$	5	10	35
7	$s-a-t$	4	5	
	$s-b-t$	5	10	50
8	$s-a-t$	4	5	
	$s-b-t$	5	10	65

and Pardolas (2010). We define the maximum ContraFlowLoc problem in static network. We also propose an efficient algorithm to solve the Problem 3.

Problem 3. Given network $N_{stat} = (V, A, u_e, s, t)$, locations L and size r_p of facility p, the maximum static ContraFlowLoc problem finds the maximum flow in $N^{\text{loc}}_{stat} = (V, A, u'_e, s, t)$ providing efficient location for the facility with arc reversals capability.

Algorithm 3: Maximum Static ContraFlowLoc

Input : A static network $N_{stat} = (V, A, u_e, s, t)$, locations L, size r_p of facility p
Output: Maximum contraflow value max_cont, location loc(p) of facility p in $N_{stat} = (V, A, u_e, s, t)$
1 Construct auxiliary $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$
2 Apply Algorithm 1 in $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ considering locations L
3 Decompose the maximum flow resulting from Step 2 into chain and cycle flows then remove the cycle flow
4 A arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u'(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$

Theorem 3. Algorithm 3 solves the maximum ContraFlowLoc problem optimally.

Proof. The Steps 1 and 2 are feasible by definition. If $r_p > u_{\overrightarrow{e}}$ and $loc(p) = \overrightarrow{e}$ on the auxiliary network then the capacity of $\overleftarrow{e}$ is defined by $u'(\overleftarrow{e}) = u(\overleftarrow{e}) + u(\overrightarrow{e}) - r_p$, since the facility will occupy all capacity of $\overrightarrow{e}$ and remaining from the capacity of $\overleftarrow{e}$. Thus, Step 4 is well defined, i.e. not both arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be reversed at a time which is ensured by using Step 3. There is flow along $\overrightarrow{e}$ or $\overleftarrow{e}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with arc reversals in $N_{stat} = (V, A, u_e, s, t)$.

Algorithm 1 has been used in Step 3 for the solution on $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$. In fact, any optimal solution to the FlowlLoc maximum flow problem with arc reversals on $N_{stat} = (V, A, u_e, s, t)$ is also a feasible solution to the maximum FlowLoc problem on $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$. Moreover, the amount of flow sent from s to t in Steps 2 is not changed in Step 4, the resulting solution is an optimal for the Problem 3. □

Corollary 2. Algorithm 3 finds optimal solution for the maximum ContraFlowLoc problem in polynomial time.

Proof. The direction of paths can be reversed using Step 3 in linear time thus the construction of auxiliary network takes linear time. Thus, the complexity of Algorithm 3 depends on the complexity of Step 2. Hence, Algorithm 3 finds optimal solution for the Problem 3 in polynomial time as designed by Theorem 3. □

Example 4. As in Example 1, let $L = \{(s, a), (s, b), (a, b), (a, t), (b, t)\}$ be the locations, nol $= 1$ and $r_p = 2$. Using contraflow approach on the same network, 15 units of flow can be send to the sink through the paths $s-a-t$ and $s-b-t$ with path flows 5 and 10, respectively. Thus, the optimal flow value is 15 with optimal location (a, b). Note that the optimal locations could be changed after using contraflow technique. However, the optimal location remains the same in this example.

Table 3: Earliest arrival FlowLoc on the network shown in Figure 1 (iii).

Locations	Time Horizon	Paths	Length	Flow	Total Earliest Arrival Flow	Remark
(s,a)	3	$s-a-b-t$	3	2	2	
	4	$s-a-b-t$ $s-a-t$	3 4	2 1	5	
	5	$s-a-t$ $s-b-t$	4 5	1 10	16	
	6	$s-a-t$ $s-b-t$	4 5	3 10	29	
	7	$s-a-t$ $s-b-t$	4 5	3 10	42	
	8	$s-a-t$ $s-b-t$	4 5	3 10	55	
(s,b)	3	$s-a-b-t$	3	2	2	
	4	$s-a-b-t$ $s-a-t$	3 4	2 3	7	
	5	$s-a-b-t$ $s-a-t$ $s-b-t$	3 4 5	2 3 8	20	
	6	$s-a-b-t$ $s-a-t$ $s-b-t$	3 4 5	2 3 8	33	
	7	$s-a-b-t$ $s-a-t$ $s-b-t$	3 4 5	2 3 8	46	
	8	$s-a-b-t$ $s-a-t$ $s-b-t$	3 4 5	2 3 8	59	
(a,t)	3	$s-a-b-t$	3	2	2	
	4	$s-a-b-t$ $s-a-t$	3 4	2 3	7	
	5	$s-a-t$ $s-b-t$	4 5	3 10	20	
	6	$s-a-t$ $s-b-t$	4 5	3 10	33	
	7	$s-a-t$ $s-b-t$	4 5	4 10	47	
	8	$s-a-t$ $s-b-t$	4 5	4 10	61	
(a,b)	4	$s-a-t$	4	5	5	
	5	$s-a-t$ $s-b-t$	4 5	5 10	20	
	6	$s-a-t$ $s-b-t$	4 5	5 10	35	
	7	$s-a-t$ $s-b-t$	4 5	5 10	50	
	8	$s-a-t$ $s-b-t$	4 5	5 10	65	Maximum but not earliest arrival

Table 4: Remaining part of Table 2.

Locations	Time Horizon	Paths	Length	Flow	Total Earliest Arrival Flow	Remark
	3	$s-a-b-t$	3	2	2	
	4	$s-a-b-t$	3	2		
		$s-a-t$	4	3	7	
	5	$s-a-t$	4	3		
		$s-b-t$	5	8	18	
(b,t)	6	$s-a-t$	4	5		
		$s-b-t$	5	8	31	
	7	$s-a-t$	4	5		
		$s-b-t$	5	8	44	
	8	$s-a-t$	4	3		
		$s-b-t$	5	10		
		$s-b-a-t$	8	2	59	

4.2 Maximum Dynamic ContraFlowLoc

The maximum dynamic ContraFlowLoc problem is introduced to locate the facility on given network such that the loss in maximum dynamic contraflow is minimum on given network, i.e., the dynamic flow on updated network is maximum. We also present an efficient algorithm for solving Problem 4.

Problem 4. Given a network $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$, locations L and size r_p of facility p, the maximum dynamic ContraFlowLoc problem on $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$ is to find the maximum dynamic flow on updated network $N^{\text{loc}}_{\text{dyna}} = (V, A, u'_e, \theta_e, T, s, t)$ with arc reversals allowed initially.

Algorithm 4: Maximum Dynamic ContraFlowLoc

Input : A dynamic network $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$, locations L, size r_p of facility p

Output: Maximum dynamic contraflow value max_dyna_cont, location $\text{loc}(p)$ of facility p in $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$

1 Construct auxiliary $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and symmetric transit time

2 Apply Algorithm 2 in $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ considering locations L

3 Decompose the maximum dynamic flow resulting from Step 3 into chain and cycle flows then remove the cycle flow

4 A arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u'(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$

Theorem 4. The maximum dynamic ContraFlowLoc problem can be solved optimally in time $O(|L|(m \log n)(m + n \log n))$.

Proof. As in Theorem 3, Steps 1, 2 and 4 are feasible and well defined. The solution on $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ has been obtained using Algorithm 2 in Step 2. Indeed, any optimal solution to the problem with arc reversals on $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$ is also a feasible solution to the problem on $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$. Every feasible flow of the maximum dynamic FlowLoc problem in $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ is feasible to the maximum dynamic ContraFlowLoc problem in $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$. The maximum dynamic contraflow with facility in $N_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ is not more than the maximum dynamic flow in $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ with facility. The maximum dynamic flow can be obtained by temporally repeating a path flow of a static graph, Ford and Fulkerson (1958). Since the amount of flow sent from s to d in Steps 3 is not changed in Step 4, the resulting FlowLoc solution is optimal for the Problem 3. As described in Corollary 2, the running time of the Algorithm 4 depends on the running time of Step 3, indeed in the running time of the Algorithm 2. Hence, the Algorithm 4 solves the Problem 4 in polynomial time, Corollary 1. □

Example 5. Apply the Algorithm 4 on the network of Figure 1 for $T = 8, r_p = 2$ and $L = \{(s,a),(s,b),(a,b),(a,t),(b,t)\}$. The optimal flow is 65 which can be send through the paths $s-a-t$and $s-b-t$ in different time intervals providing optimal location (a, b) (see Table 2 corresponding to location (a, b)). Here, the optimal flow value after contraflow is more than the double flow value before contraflow configuration.

The earliest arrival flow model has been integrated with contraflow to introduce earliest arrival contraflow by Dhungana, Pyakurel, Khadka, and Dhamala (2015). They have presented different algorithms to get exact and approximate solutions. According to the idea of FlowLoc presented in Section 3.2 and Example 3 we are not combining the earliest arrival contraflow model with location analysis to define the FlowLoc earliest arrival contraflow problem.

5. CONTRAFLOWLOC PROBLEMS FOR EXTENDED LOCATIONS

The FlowLoc problems with given possible facility locations are studied in Section 4. In this section we consider a FlowLoc problem, where a location will be feasible if its size is at least the respective arc capacity. The problem is modeled with and without the lane reversal permissibility. In both cases, efficient algorithms are presented. In this approach, the contraflow configuration increases the number of facilities with modified arc capacities.

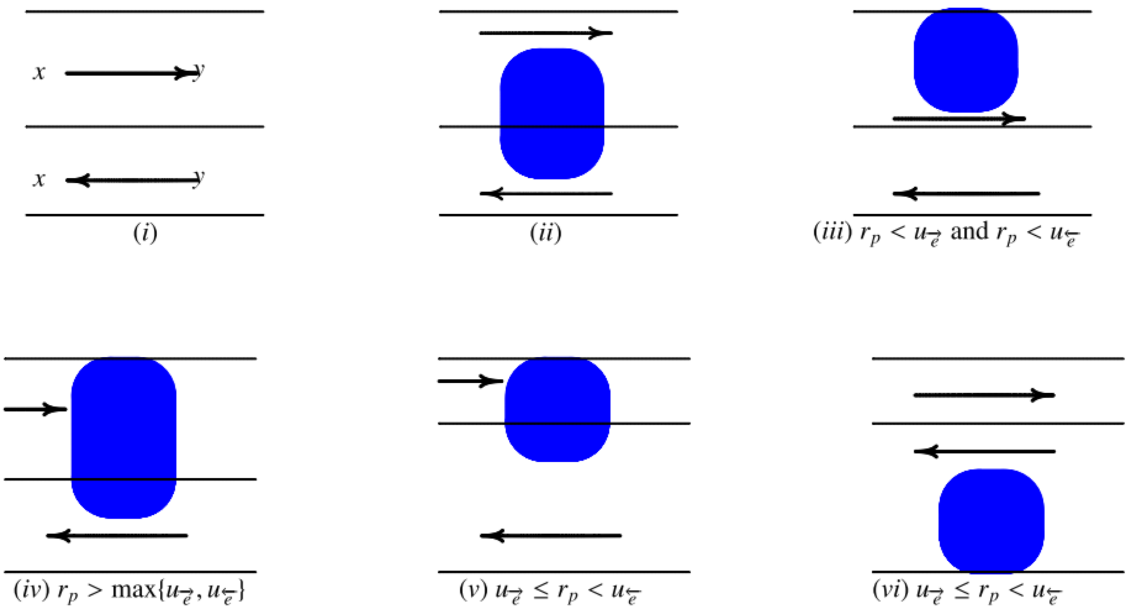


Figure 2: Variations of facility locations on auxiliary network with respect to the modified capacities of arcs and facility locations determined by a given facility size.

Consider Figure 2(i), which represents two way arcs (x, y) and (y, x). The maximum ContraFlowLoc solution locates a facility on the auxiliary network and finds an optimal flow in the network with reduced capacity by r_p. We exclude of locating a facility on the middle of a road as shown in Figure 2(iii). We consider the following cases.

1. Suppose the facility is located on (x, y) as shown in Figure 2(ii). If a maximum contraflow is in the direction of (x, y) and it is greater than $u'(x, y)$, then the direction of arc (y, x) has to be reversed.
2. Suppose the capacity of facility is greater than the capacities of both arcs (x, y) and (y, x), then the arc reversal problem reduces to the single arc problem. In this situation, the facility can be located on any side, for example as shown in Figure 2(iv).
3. Suppose the capacity of only one arc is sufficient to allocate facility but not its backward arc, then the problem becomes as shown in Figures 2(v) and 2(vi). The former is as in Figure 2(iv) and the latter is as in Figure 2(ii).

Example 6. Let $r_p = 3$, $u(a, b) = 2$ and $u(b, a) = 4$ on a single arc of a network. Suppose that we locate facility on (a, b) in the auxiliary network and a maximum-flow solution in the auxiliary network yields a flow 2 along the direction of (a, b). Then, the arc (b, a) should be reversed. But if we would have located the facility on (b, a) in the auxiliary network then there was no need to reverse the above arc to pass this flow value. it matches to the case of Figure 2 (v-vi) . Indicating that a wrong decision may lead to unnecessary arc reversals.

5.1 Maximum Static ContraFlowLoc

The maximum static ContraFlowLoc problem and its solution procedure has been presented in this section.

Problem 5. Given network $N_{stat} = (V, A, u_e, s, t)$, locations $L = \{e \in A \mid u_e \geq r_p\}$, where r_p denotes the size of facility p, the maximum static ContraFlowLoc problem is to find the maximum flow in $N_{stat}^{\text{loc}} = (V, A, u'_e, s, t)$ with arc reversal capability.

Algorithm 5: Maximum Static ContraFlowLoc with Extended Locations

Input : A static network $N_{stat} = (V, A, u_e, s, t)$, locations $L = \{e \in A \mid u_e \geq r_p\}$ where r_p denotes the size of facility p

Output: Maximum contraflow value max_cont, location loc(p) of facility p in $N_{stat} = (V, A, u_e, s, t)$

1 Construct auxiliary network $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ with new capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and locations $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$

2 Apply Algorithm 1 in $\tilde{N}_{stat} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, s, t)$ considering modified set of locations $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$

3 Decompose the maximum flow resulting from Step 2 into chain and cycle flows then remove the cycle flow

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$

5 If facility is located on $\tilde{e} \in \tilde{A}$ and the flow is positive along the direction of the arc $\overrightarrow{e} \in A$ then allocate the facility along the direction of $\overleftarrow{e} \in A$

Theorem 5. The maximum ContraFlowLoc problem with extended locations can be solved optimally in time $O(|\tilde{L}|nm)$.

Proof. The construction of possible location set $L = \{e \in A \mid u_e \geq r_p\}$ is feasible as the facility will host in any location if it has sufficient capacity. Step 1 constructs the auxiliary network with new location set $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$ and capacity $\tilde{u}$ that are feasible. Step 2 is feasible as it calculates FlowLoc solution in the auxiliary network considering location $\tilde{L}$. Step 4 is well defined, i.e. not both arcs $\overrightarrow{e}$ and $\overleftarrow{e}$ have to be reversed at a time, this is ensured by the solution in auxiliary network. The resulting flow is decomposed into path and cycle flows and the cycle flows are removed in Step 3. So that there is flow along $\overrightarrow{e}$ or $\overleftarrow{e}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with arc reversals in $N_{stat} = (V, A, u_e, s, t)$.

Optimal locations for the facility will be obtained in Step 2 for auxiliary network. If there are two way arcs on given network and the auxiliary arcs host the facility on auxiliary network then Step 5 decides the location of the facility in the given network according to the direction of flow provided from Step 3. Thus, Step 5 is feasible. Further proof of this theorem can be completed as in Theorem 3. □

Example 7. Let nol $= 1$ and $r_p = 4$ with ignored travel times in Figure 1 (i). Then, $L = \{(a, t), (s, b), (b, s), (b, t), (t, b)\}$ be the possible locations. Applying Algorithm 1, we can send 7 units of flow before contraflow. In this case, both locations (b, s) or $(t, b)\}$ are the optimal location as the resulting flow remains the same. The arc (b, a) or (a, b) was not able to host the facility due to low capacities.

But using contraflow approach, both arcs have been combined to increase the capacity. As a result, all arcs become feasible to host the facility with capacity 4. After locating facility on the arc, 15 units of flow can be send to the sink through the paths $s - a - t$ and $s - b - t$ with path flows 5 and 10, respectively. Thus, the optimal flow value is 15 with optimal location (a, b). Here, the facility on the network did not affect the optimality of the flow.

5.2 Maximum Dynamic ContraFlowLoc

The maximum dynamic ContraFlowLoc problem with capacity constrained locations and its procedure has been introduced in this section.

Problem 6. Given a network $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$, locations $L = \{e \in A \mid u_e \geq r_p\}$, where r_p denotes the size of facility p, the maximum dynamic ContraFlowLoc problem on $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$ is to find the maximum dynamic flow on updated network $N_{\text{dyna}}^{\text{loc}} = (V, A, u'_e, \theta_e, T, s, t)$ with arc reversals allowed initially.

Theorem 6. Algorithm 6 solves Problem 4 optimally in time $O(|\tilde{L}|(m \log n)(m + n \log n))$.

Proof. The feasibility of Algorithm 6 can be proved as in Theorem 5. The solutions on $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ with locations $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$ have been calculated using Algorithm 2 in Step 2. The maximum dynamic flow obtained from Step 2 has been decomposed into temporally repeated path flows in Step 3 which is optimal flow for the Problem 6. A proof can be shown as in Theorem 4. □

Example 8. Consider the network of Figure 3 with values from Table 5 and $T = 9$. Without providing facility on network, maximum 61 and 137 units of flows can be send before and after contraflow, respectively. Our objective is to locate a facility on arc with minimum flow loss in the given network. Suppose nol $= 1$, $r_p = 10$ and $T = 9$. Before contraflow, possible locations are $L = \{(s,a),(s,d),(c,t)\}$. Construct the auxiliary network (shown in Figure 4) with values of Table 6. Then, possible locations set becomes $\tilde{L} = \{(s,a),(s,d),(c,t),(d,a),(d,e),(b,t)\}$. Algorithm 6 locates facility on given evacuation network of Figure 3 with minimum flow loss. Different maximum dynamic flow solutions after providing facility on different locations with different time horizons are shown in Table 7. Before contraflow, the arc (d,a) or (a,d) was not feasible to host the facility due to low capacities. But using contraflow approach, arc capacity has been increased by reversing the direction. As a result, the arc become feasible to host the facility and it is optimal location for the problem. If we consider different time horizon less than the transit time of longest path (longest in the sense of transit time) optimal location can be different for different time horizons, details is shown in Table 7.

Algorithm 6: Maximum Dynamic ContraFlowLoc with Extended Locations

Input : A dynamic network $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$, locations $L = \{e \in A \mid u_e \geq r_p\}$, size r_p of facility p

Output: Maximum dynamic contraflow value max$_$dyna$_$cont, location loc(p) of facility p in $N_{\text{dyna}} = (V, A, u_e, \theta_e, T, s, t)$

1 Construct auxiliary $\tilde{N}^{\text{loc}}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ with new locations $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$, capacity $\tilde{u}(\tilde{e}) = u(\overrightarrow{e}) + u(\overleftarrow{e})$ and symmetric transit time $\tilde{\theta}$

2 Apply Algorithm 2 in $\tilde{N}_{\text{dyna}} = (V, \tilde{A}, \tilde{u}_{\tilde{e}}, \tilde{\theta}_{\tilde{e}}, T, s, t)$ for locations $\tilde{L} = \{\tilde{e} \in \tilde{A} \mid \tilde{u}_{\tilde{e}} \geq r_p\}$

3 Construct temporally repeated flow resulting from Step 2

4 An arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $u(\overrightarrow{e})$ or there is a non-negative flow along the arc $\overrightarrow{e} \notin A$

5 If facility is located on $\tilde{e} \in \tilde{A}$ and the flow is positive along the direction of the arc $\overrightarrow{e} \in A$ then allocate the facility along the direction of $\overleftarrow{e} \in A$

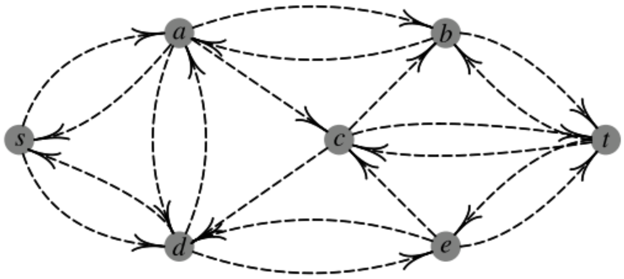


Figure 3: Evacuation network.

Table 5: Capacities and transit times of arcs corresponding to Figure 3.

Arcs	(s,a)	(a, s)	(s, d)	(d, s)	(d, a)	(a,d)	(a, c)	(c, d)	(a, b)	(b, a)
Capacities	10	5	10	7	7	4	3	5	5	3
Transit times	1	1	2	2	1	1	2	2	1	1
Arcs	(c, b)	(e, c)	(d, e)	(e, d)	(c, t)	(t, c)	(b, t)	(t, b)	(e, t)	(t, e)
Capacities	6	5	8	3	10	5	2	9	4	4
Transit times	1	1	2	2	1	1	1	1	3	3

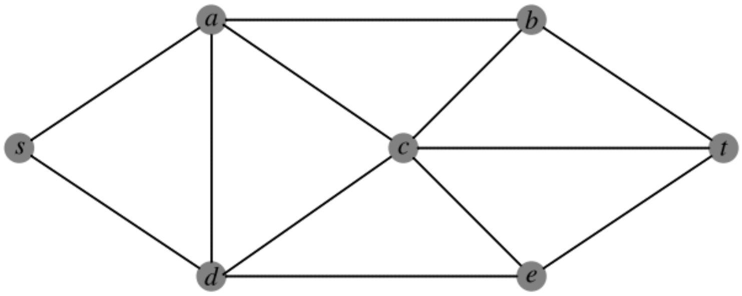


Figure 4: Auxiliary network corresponding to Figure 3.

Table 6: Capacities and transit times of arcs corresponding to Figure 4.

Auxiliary Arcs	(s,a)	(s, d)	(d, a)	(a, c)	(c, b)	(d, c)	(a, b)	(e, c)	(d, e)	(t, c)	(b, t)	(e, t)
Capacities	15	17	11	3	6	5	8	5	11	15	11	8
Transit times	1	2	1	2	1	2	1	1	2	1	1	3

Table 7: Variants of FlowLoc solutions for Network 4 considering the parameters of Table 5.

T	Before contraflow				After contraflow							Difference of Two Optimal Flows
	MDF after placing facility on			Location decisions	Maximum dynamic flow(MDF) after placing facility on						Location decisions	
	(s, a)	(s, d)	(c, t)		(s, a)	(s, d)	(c, t)	(d, a)	(d, e)	(b, t)		
3	0	2	2	(s, d) or (c, t)	5	8	8	8	8	1	(s, d) or (c, t) (d, a) or (d, e)	6
4	0	7	4	(s, d)	10	19	19	19	19	5	(s, d) or (c, t) (d, a) or (d, e)	11
5	2	12	6	(s, d)	23	35	32	31	35	14	(s, d) or (d, e)	23
6	12	21	8	(s, d)	44	56	48	52	53	28	(s, d)	35
7	22	30	14	(s, d)	66	78	72	79	70	48	(d, a)	49
8	32	39	20	(s, d)	88	100	96	106	87	68	(d, a)	67
9	42	48	26	(s, d)	110	122	120	133	104	88	(d, a)	85

6. CONCLUSIONS

In this paper, we studied the maximum FlowLoc and contraflow models on both static and dynamic networks. The maximum FlowLoc model locates facility in the possible locations and maximize the flow on the updated network whereas the contraflow approach reverses the direction of arcs to increase the flow value.

We introduced the ContraFlowLoc model that locates facility on given network and maximize the flow on updated network where arcs can be revered if the optimal solution can be improved from the given network. The maximum FlowLoc over continuous time, maximum static ContraFlowLoc and maximum dynamic ContraFlowLoc problems are studied in this paper. We also proposed efficient algorithms for the maximum dynamic FlowLoc, maximum ContraFlowLoc and maximum dynamic ContraFlowLoc in two-terminal networks for given possible locations as well as extended locations.

The extension of ContraFlowLoc problem to achieve different objectives such as lexicographically maximum ContraFlowLoc, lexicographically maximum dynamic ContraFlowLoc, earliest arrival transshipment ContraFlowLoc and quickest ContraFlowLoc for different networks with both given and arbitrary locations could be the further research interest.

ACKNOWLEDGEMENT

Authors acknowledge the partial supports of DAAD Partnership Program and AvH Research Group Linkage Program. The first author also acknowledges University Grants Commission, Nepal for a support in this research.

REFERENCES

Daskin, M. (1995). *Network and discrete location: Models, algorithms, and applications*. New York: Wiley Interscience.

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

Dhungana, R. C., Pyakurel, U., & Dhamala, T. N. (2018). Abstract contraflow models and solution procedures for evacuation planning. *Journal of Mathematics Research*, *10*(4), 89-100.

Dhungana, R. C., Pyakurel, U., Khadka, S. R., & Dhamala, T. N. (2015). Universally maximum contraflow on for evacuation planning. *International Journal of Operations Research Nepal*, *4*(1), 67-78.

Drezner, Z., & Hamacher, H. (2002). *Facility location: Applications and theory*. Berlin: Springer.

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

Ford, L. R., & Fulkerson, D. R. (1958). Constructing maximal dynamic flows from static network. *Operations Research*, *6*, 416-433.

Hamacher, H. W., Heller, S., & Rupp, B. (2013). Flow location (flowloc) problems: dynamic network flows and location models for evacuation planning. *Annals of Operation Research*, *207*, 161-180.

Heller, S., & Hamacher, H. (2011). The multi terminal q-flowloc problem: A heuristic. In *Proceedings of the international network optimization conference (INOC)* (p. 523-528). Berlin: Springer.

Hua, J., Ren, G., Cheng, Y., & Ran, B. (2014). An integrated contraflow strategy for multi modal evacuation. *Hindawi Publishing Corporation Mathematical Problems in Engineering*.

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

Orlin, J. B. (2013). Max flows in $o(nm)$ time, or better. In *Proceedings of forty-fifth annual acm symposium on theory of computing* (p. 765-774).

Philpott, A. (1982). *Algorithms for continuous network flow problems* (Ph.D. Thesis). University of Cambridge, UK.

Pyakurel, U., & Dhamala, T. N. (2016). Continuous time dynamic contraflow models and algorithms. *Hindawi Publishing Corporation Advances in Operations Research,* Volume 2016, Article ID 7902460.

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

Pyakurel, U., Nath, H. N., & Dhamala, T. N. (2018a). Efficient contraflow algorithms for quickest evacuation planning. *Science China Mathematics*. doi: 10.1007/s11425-017-9264-3

Pyakurel, U., Nath, H. N., & Dhamala, T. N. (2018b). Partial contraflow with path reversals for evacuation planning. *Annals of Operation Research*. doi: 10.1007/s10479-018-3031-8

Rebennack, S., Arulselvan, A., Elefteriadou, L., & Pardolas. (2010). Complexity analysis of maximum flow problem with arc reversal. *Journal of Combinatorial Optimization*, *29*, 200-216.

Weber, A. (1909). *Über den Standort der Industrien, Tübingen. Theory of the Location of Industries*. University of Chicago Press (English translation by C. J. Friedrich (1929)).

Wilkinson, W. L. (1971). An algorithm for universal maximal dynamic flows in a network. *Operations Research*, *19(7)*, 1602-1612.

Xie, C., Lin, D. Y., & Waller, S. (2010). A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E*, *46*, 295-316.

Xie, C., & Turnquist, M. A. (2011). Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. *Transportation Research Part C*, *19*, 40-63.

Zhao, X., Feng, Z.-Y., Li, Y., & Bernard, A. (2016). Evacuation network optimization model with lane-based reversal and routing. *Hindawi Publishing Corporation Advances in Operations Research,* Volume 2016, Article ID 1273508.

Journal of Mathematics Research; Vol. 10, No. 4; August 2018
ISSN 1916-9795 E-ISSN 1916-9809
Published by Canadian Center of Science and Education

Abstract Contraflow Models and Solution Procedures for Evacuation Planning

Ram Chandra Dhungana[1], Urmila Pyakurel[1] & Tanka Nath Dhamala[1]

[1] Central Department of Mathematics, Tribhuvan University, P.O.Box 13143, Kathmandu, Nepal

Correspondence: Tanka Nath Dhamala, Central Department of Mathematics, Tribhuvan University, P.O.Box 13143, Kathmandu, Nepal.

Received: April 28, 2018 Accepted: May 14, 2018 Online Published: June 8, 2018

doi:10.5539/jmr.v10n4p89 URL: https://doi.org/10.5539/jmr.v10n4p89

Abstract

The abstract flow model deals with the flow paths (routes) that satisfy the switching property. Contraflow is a widely accepted solution approach that increases the flow and decreases the evacuation time making the traffic smooth during evacuation by reversing the required road directions from the risk areas to the safe places. In this paper, we integrate the concepts of abstract flow and contraflow, give mathematical formulations of these models and present efficient algorithms for solving the abstract contraflow problems. The efficient solution procedures are presented for maximum dynamic, lexicographically maximum and earliest arrival abstract contraflow problems. This approach maximizes the flow value in given time and seeks to eliminate the crossing conflicts.

Keywords: Abstract flow, evacuation planning, contraflow, maximum flow, earliest arrival flow

MSC(2010) Primary: 90B10, 90C27, 68Q25; Secondary: 90B06, 90B20.

1. Introduction

Due to different disasters, challenges of emergency management have been increased day by day. It is necessary to evacuate as many people as possible within the limited time horizon under dynamic hazard from the dangerous states (sources) to the safe places (sinks). For the effective evacuation planning different issues such as identifying transportation routes, notifying evacuees, common traffic delays, etc, should be managed. During the evacuation, the prominent dynamic network flow models have been widely used to find out the efficient evacuation routes that were firstly introduced and investigated by (Ford & Fulkerson, 1958). There exist different models and algorithms for buildings, stadiums, ships, districts, cities or whole sub-national region evacuation based on their scenarios. For different variants of the dynamic flow problems and their corresponding results, we refer to (Altay & Green III, 2006; Dhamala, 2015; Hamacher & Tjandra, 2001; Jarvis & Ratliff, 1982; Moriarty *et al.*, 2007; Pel *et al.*, 2012; Schadschneider *et al.*, 2009). The evacuation network is interpreted by a directed graph where the intersections of roads (i.e., rooms in a building or intersection of streets in a region) are represented by nodes, road segments between nodes (i.e., doors between rooms, or streets in region) are represented by arcs and routes are taken as paths. The places where evacuees are situated and start to move are considered as source nodes and the safe places where they are supposed to arrive are destination (sink) nodes. Each node and arc has a non-negative integer capacity, where the capacity gives the maximum number of evacuees allowed at the element.

After the development of maximal static flow and maximum dynamic flow models and algorithms by (Ford & Fulkerson, 1956; Ford & Fulkerson, 1958), the existence of lexicographically maximal (lex-max) flow was shown by (Minieka, 1973) in the time expanded network for multiterminal problems. For the single source multi-sink lex-max problem with the restriction that the source should be ranked first without pre-specified ordering of the sinks, (Megiddo, 1974) presented a solution procedure that seeks a maximum flow out of sources while maximizing the minimum flow entering any sink and any pair of sinks. Both procedures have pseudo-polynomial time complexity. A polynomial time algorithm to solve the general lex-max dynamic flow problem on the original graph by using chain decomposable flows has been presented in (Hoppe & Tardos, 1994; Hoppe & Tardos, 2000).

The abstract flows introduced in (Hoffman, 1974) generalizes the concept of paths by replacing the underlying network configuration. The maximum dynamic abstract flow problem and its solution procedure has been investigated by (Kappmeier *et al.*, 2014). The lexicographically maximum abstract flow problem has been investigated by (Kappmeier, 2015). This approach makes the use of, so-called switching property that eliminates the crossing at intersections. Some of the important lane-based routing strategy for reducing the delays that reduce (or eliminate) crossing and merging conflicts at intersections have been studied in (Zhao *et al.*, 2016). In a lane-based routing plan, selecting and turning options at intersections are restricted to improve traffic flow away from a hazardous area. Intersections with potentially significant delays can be temporarily transformed into an uninterrupted flow facility which is the most beneficial aspect of this routing.

Contraflow approach is another emerging and widely accepted model for evacuation planning. It increases the outbound road capacities by reversing the direction of roads towards the safe destinations. (Kim *et al.*, 2008) give first integer programming formulation and presented different heuristic solutions for large scale evacuations, where the evacuation time has been improved by at least 40 percent with at most 30 percent of the total arc reversals. They showed that the problem of minimizing the evacuation time is $\mathcal{NP}$-hard. The application of contraflow is not only limited to evacuation planning but also in traffic planning that reduces congestion and traffic jams during the day-to-day office hours, some accident management cases or some street exhibitions. Various mathematical models, heuristics, optimization and simulation techniques taking into account of macroscopic and microscopic behavioral characteristics deal with contraflow for this transportation network, however an acceptable contraflow solution even approximately is a lacking due to very high computational costs.

The maximum dynamic contraflow problem has been introduced in (Arulselvan, 2009; Rebennack *et al.*, 2010) and solved with polynomial algorithm for two terminal general network. They proved that the problem is $\mathcal{NP}$-hard for multi-terminal network. Authors in (Dhamala & Pyakurel, 2013; Dhungana *et al.*, 2015; Pyakurel & Dhamala, 2015; Pyakurel & D-hamala, 2017a; Pyakurel & Dhamala, 2016) introduced the earliest arrival, quickest transshipment and lex-maximum dynamic contraflow problems in both discrete and continuous time settings. The former is solved in strongly polynomial and pseudo-polynomial time on two terminal series-parallel and general networks, respectively. They presented polynomial algorithm for the quickest transshipment contraflow problem. For the given priority ordering, the lex-maximum dynamic contraflow problem is solved in polynomial time for multi-terminal network. For the fixed the supply and demands, the earliest arrival transshipment contraflow has been introduced in (Pyakurel & Dhamala, 2017a; Pyakurel & Dhamala, 2017b) in discrete and continuous times. They solved the problem in multi-source and single sink network as well as single source and multi-sink network. In both cases, their algorithms have polynomial time complexity. Moreover, they presented approximation algorithms to compute the approximate earliest arrival transshipment contraflow for multi-terminal networks, (Pyakurel & Dhamala, 2017b; Pyakurel *et al.*, 2017). The contraflow approach is generalized on lossy network in (Pyakurel *et al.*, 2014). They solved the generalized maximum dynamic and generalized earliest arrival contraflow problems on two terminal lossy network. For more details, we refer to (Pyakurel, 2016). Moreover, authors in (Pyakurel *et al.*, 2017) introduced the abstract contraflow approach with path reversal capability. They presented a polynomial time algorithm to solve the abstract maximum dynamic contraflow in continuous time setting. The quickest contraflow problems with constant and load dependent transit times are solved with computational experiments, (Pyakurel *et al.*, 2018).

Authors in (Xie & Turnquist, 2011) solved the lane based contraflow and crossing elimination strategies at intersections jointly. A network optimization model to integrate these problems has been introduced in (Zhao *et al.*, 2016). The bi-level lane-based network optimization and simulation model have been formulated in (Xie *et al.*, 2010), where the upper level optimizes the network evacuation performance subject to the contraflow and crossing-elimination constraints, and the lower level simulates dynamic evacuation flows. To deal the uncertain arrivals of evacuees with low mobility population, the multi-model integrated contraflow has been presented in (Hua *et al.*, 2014). It contains non-contraflow to shorten the strategy setup time, full-lane contraflow to minimize the evacuation network capacity and bus contraflow to realize the transit cycle operation. A more realistic contraflow problem with evacuation priorities and the setup time has been considered in (Wang *et al.*, 2013).

In this work, we introduce and investigate abstract contraflow models by combining the concepts of abstract flow and contraflow approaches. We present some efficient solution procedures to maximize the achievable number of evacuees at minimum time by reducing the possible crossing conflicts. Our contributions resemble the results in contraflow on non-abstract networks presented by the different authors (Pyakurel & Dhamala, 2015; Pyakurel & Dhamala, 2016; Rebennack *et al.*, 2010). Throughout the paper, we reverse the paths to increase the path capacities whenever it is necessary by applying the concepts of (Rebennack *et al.*, 2010) and (Pyakurel *et al.*, 2017). Then the problems are solved by using abstract flow algorithms of (Kappmeier *et al.*, 2014) and (Kappmeier, 2015).

The structure of the paper is as follows. The notations and prior works in network flows are presented in Section 2. We propose maximum dynamic and earliest arrival abstract contraflow models with efficient solution procedures for two-terminal abstract networks in Section 3. The lexicographically maximum contraflow model and algorithm for multiterminal abstract networks are presented in Section 3. The paper concludes with Section 4.

2. Some of the Basic Notations and Models

Let $N = (E, \Gamma, b, \tau, S, D, T)$ be a multi-terminal evacuation network, where E and Γ represent the sets of elements (also called ground set) and paths, respectively. The nonnegative capacity and travel time vectors of $e \in E$ are $b_e \in R_+^E$ and $\tau_e \in Z_+^E$, respectively. The given nonnegative integer time horizon T may be represented as discrete time periods

$\mathbf{T} = \{0, 1, \cdots, T\}$. A network N is called abstract if for every $\gamma \in \Gamma$ there is a linear order $<_\gamma$ of elements and the switching property is satisfied in Γ. A switching property requires that for every $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, there exist paths $\gamma_1 \times_e \gamma_2 \subseteq \{a \in \gamma_1 : a \leqslant_{\gamma_1} e\} \cup \{a \in \gamma_2 : a \geqslant_{\gamma_2} e\}$ and $\gamma_2 \times_e \gamma_1 \subseteq \{a \in \gamma_2 : a \leqslant_{\gamma_2} e\} \cup \{a \in \gamma_1 : a \geqslant_{\gamma_1} e\}$. We denote the parts of the path γ before and after an element e excluding it as $\gamma_{\rightarrow e} = (\gamma, e) = \{p \in \gamma : p <_\gamma e\}$ and $\gamma_{e \rightarrow} = (e, \gamma) = \{p \in \gamma : p >_\gamma e\}$. A network is considered as static or dynamic depending on time.

Example 1. *Let* $E = \{s, a, b, c, d, e, z\}$ *and* $\Gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6\}$*, where* $\gamma_1 = \{s, a, c, z\}, \gamma_2 = \{s, b, d, z\}, \gamma_3 = \{s, a, e, d, z\}, \gamma_4 = \{s, b, e, c, z\}, \gamma_5 = \{s, a, e, c, z\}, \gamma_6 = \{s, b, e, d, z\}$*. The path system satisfies the switching property with defined linear order on each path and* (E, Γ) *is an abstract network (cf. Figure 1 (iii)). For example,* $\gamma_3 \times_e \gamma_4 = \{s, a, e, c, z\}$ *and* $\gamma_4 \times_e \gamma_3 = \{s, b, e, d, z\}$ *both exist in* Γ*.*

The maximum static abstract flow problem is to assign the nonnegative flow values $f(\gamma) \in R_+^\Gamma$ to the paths so that the total flow value is maximized and the element capacity restrictions are respected. The problem can be generalized by introducing a weight function $\omega \in R_+^\Gamma$ that specifies the reward per unit of flow sent along each path. As the weights make the general problem $\mathcal{NP}$-hard, (Hoffman, 1974), their choice is restricted to supermodular functions, i.e., $\omega(\gamma_1 \times_e \gamma_2) + \omega(\gamma_2 \times_e \gamma_1) \geq \omega(\gamma_1) + \omega(\gamma_2)$ for every $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$.

For the static case, the generalized maximum weighted abstract flow and the generalized minimum weighted abstract cut problems are primal (1) and dual (2) LP problems to each other, (Hoffman, 1974).

$$\max\{\sum_{\gamma \in \Gamma} \omega(\gamma) f(\gamma) \mid \sum_{\gamma \in \Gamma : e \in \gamma} f(\gamma) \leq b_e, f(\gamma) \geq 0, \ \forall \ \gamma \in \Gamma, e \in E\} \tag{1}$$

$$\min\{\sum_{e \in E} b_e y(e) \mid \sum_{e \in \gamma} y(e) \geq \omega(\gamma), y(e) \geq 0, \ \forall \ \gamma \in \Gamma, e \in E\} \tag{2}$$

where a value $y(e)$ is assigned to every element $e \in E$ covering every path according to its weight.

The dynamic abstract flow (DAF) problems are dealt with the corresponding time expended networks. (Kappmeier *et al.*, 2014) introduced the holdover of flow at intermediate nodes to construct an abstract time expanded network (ATEN). Let $\sigma : \Gamma \rightarrow \{1, 2, \cdots, T\}$ be waiting periods for each elements of Γ. Flow enters to $e \in \gamma$ at time $\sum_{q \in \gamma_{\rightarrow e}} (\sigma_q + \tau_q) + \tau_e$ as it traveling along γ waits σ_q time units before passing through e. The set $\gamma^\sigma = \{e^\kappa \in E_T = E \times T | e \in \gamma, \sum_{q \in \gamma_{\rightarrow e}} (\sigma_q + \tau_q) + \tau_e = \kappa\}$ represents temporal paths with intermediate waiting and satisfy $q^\kappa <_{\gamma^\sigma} e^{\kappa'}$ if and only if $q <_\gamma e$. Such paths that arrive within T is $\Gamma_T^\sigma = \{\gamma^\sigma | \gamma \in \Gamma, \sigma \in \{1, 2, \cdots, T\}^\Gamma, \sum_{e \in \gamma} \{\sigma_e + \tau_e\} \leq T\}$. The set (E_T, Γ_T^σ) represent the ATEN of (E, Γ).

The maximum dynamic flow problem or the $\mathcal{NP}$-hard minimum cost dynamic flow problem have optimal solutions with local flow conservation at each intermediate elements (Fleischer & Skutella, 2003; Ford & Fulkerson, 1958). For the DAF problem, the waiting has no influence on the optimality and the temporally repeated optimal solutions can be obtained even if waiting is allowed, (Kappmeier *et al.*, 2014). The DAF is a function $f_{dyn} : \Gamma_T^\sigma \rightarrow R_+$ which is feasible if and only if the capacity of every element at every point of time is satisfied. The maximum dynamic abstract flow (MDAF) problem (3) is to maximize the total flow value respecting the given restrictions, (Kappmeier *et al.*, 2014).

$$\max \left\{ \sum_{\gamma_t \in \Gamma_T^\sigma} f_{dyn}(\gamma_t) \mid \sum_{\gamma_t \in \Gamma_T^\sigma : (e, \theta) \in \gamma_t} f_{dyn}(\gamma_t) \leq b_e, f_{dyn}(\gamma_t) \geq 0, \ \forall \ \gamma_t \in \Gamma_T^\sigma, e \in E, \theta \in \mathbf{T} \right\} \tag{3}$$

The dynamic cut is $C_{dyn} = \{(e, \theta) \in E_T : \alpha(e) \leq \theta < \alpha(e) + \tilde{y}(e)\}$, where $\tilde{y}$ is the static weighted abstract dual integral optimal solution with weight $\omega(\gamma)$ and $\alpha(e) := \min_{\gamma \in \Gamma} \sum_{\gamma \in (\Gamma, e)} (\tau(\gamma) + \tilde{y}(\gamma))$.

The outflow and inflow of a source and a sink are $|f|_s^+ = \sum_{\gamma \in \delta_s^+} f_\gamma$, and $|f|_z^- = \sum_{\gamma \in \delta_z^-} f_\gamma$, respectively. Let f^1 and f^2 be maximum abstract flows, and let $s_1, s_2, \cdots, s_k$ and $z_1, z_2, \cdots, z_k$ be the orders of sources and sinks, respectively. We say that f^2 is lexicographically smaller than f^1, denoted by $f^1 \geq_L f^2$, if there exists either an $l \in \{0, 1, \cdots, k-1\}$ such that $|f^1|_{s_{l+1}}^+ > |f^2|_{s_{l+1}}^+$ and $|f^1|_{s_i}^+ = |f^2|_{s_i}^+$ for $i = 1, 2, \cdots, l$, or all $|f^1|_{s_i}^+ = |f^2|_{s_i}^+$ for $i = 1, \ldots, k$. Similarly, $f^1 \geq_L f^2$ if either $|f^1|_{z_{l+1}}^- > |f^2|_{z_{l+1}}^-$ and $|f^1|_{z_i}^- = |f^2|_{z_i}^-$ for some $l \in \{0, 1, \cdots, k-1\}$ and $i = 1, 2, \cdots, l$ or $|f^1|_{z_i}^- = |f^2|_{z_i}^-$ for all $i = 1, \cdots, k$. The lexicographically maximum abstract flow (LMAF) f^* is a maximum abstract flow (MAF) respecting the terminal orders, i.e. $f^* \geq_L f$ for all abstract flows f.

A sequence of terminals is compatible if the terminal elements respect their rank for more than one terminals of the same type appeared on a path. For sources $s_1, s_2, \cdots, s_k$, it holds that $\gamma \in \Gamma, s_i \neq s_j \in \gamma : j < i \Rightarrow s_i \leq_\gamma s_j$. But a sequence of sinks $z_1, z_2, \cdots, z_k$ has to assure $\gamma \in \Gamma, z_i \neq z_j \in \gamma : i < j \Rightarrow z_i \leq_\gamma z_j$. For given a compatible sequence of sources and sinks, we define abstract networks (E, Γ_s^i) with increasing subsets of paths $\Gamma_s^i \subset \Gamma$ for $i = 1, 2, \cdots, k$, where $\Gamma_s^i =$

$\Gamma_s^{i-1} \cup \{\gamma \in \Gamma | s_i = \text{first}(\gamma)\}$ with $\Gamma_s^0 = \emptyset$. Similarly, we define (E, Γ_z^i) with paths $\Gamma_z^i \subset \Gamma$, where $\Gamma_z^i = \Gamma_z^{i-1} \cup \{\gamma \in \Gamma | z_i = \text{last}(\gamma)\}$ with $\Gamma_z^0 = \emptyset$. Each of (E, Γ_s^i) and (E, Γ_z^i) contains the paths starting and ending in the first i sources and sinks, respectively.

The central idea behind the contraflow technique is to improve the outbound capacity by adopting the arc or path reversals toward the safer places keeping the same travel time in the evacuation network. As a result the flow value is increased, evacuation time is decreased and traffic flow is made smooth. Let $\Gamma = \{\overrightarrow{\gamma}, \overleftarrow{\gamma}\}$ be the set of all paths in contraflow abstract network N with capacities $b(\overrightarrow{\gamma}) = \min\{b_e : e \in \overrightarrow{\gamma}\}$ and $b(\overleftarrow{\gamma}) = \min\{b_e : e \in \overleftarrow{\gamma}\}$. We define the undirected auxiliary network $\tilde{N}$ by adding the capacities on the corresponding two-way paths and keeping the transit time (if any) fixed. The set of elements and paths are denoted by $\tilde{E}$ and $\tilde{\Gamma}$, where $\tilde{e} \in \tilde{E}$ and $\tilde{\gamma} \in \tilde{\Gamma}$. Then the capacity function is defined as $b(\tilde{\gamma}) = \min\{b_{\tilde{e}} : \tilde{e} \in \tilde{\gamma}\}$ while the travel times (if any) on paths remains the same.

By construction of auxiliary network with path reversal, $\tilde{\Gamma}$ satisfies the switching property and the order of elements holds for each $\tilde{\gamma} \in \tilde{\Gamma}$. This implies that the auxiliary network of the abstract evacuation network is also an abstract network.

3. Abstract Contraflow Problems

In this section, we define abstract contraflow problems and present efficient solution procedures to solve them. This approach uses path reversals in abstract network at time zero without any switching costs.

Example 2. *Let* $E = \{s, a, b, c, d, e, z\}$ *and* $\Gamma = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overrightarrow{\gamma_3}, \overrightarrow{\gamma_4}, \overrightarrow{\gamma_5}, \overrightarrow{\gamma_6}, \overleftarrow{\gamma_1}, \overleftarrow{\gamma_2}, \overleftarrow{\gamma_3}, \overleftarrow{\gamma_4}, \overleftarrow{\gamma_5}, \overleftarrow{\gamma_6}\}$ *with* $\overrightarrow{\gamma_1} = \{s, a, c, z\}$, $\overrightarrow{\gamma_2} = \{s, b, d, z\}$, $\overrightarrow{\gamma_3} = \{s, a, e, d, z\}$, $\overrightarrow{\gamma_4} = \{s, b, e, c, z\}$, $\overrightarrow{\gamma_5} = \{s, a, e, c, z\}$, $\overrightarrow{\gamma_6} = \{s, b, e, d, z\}$, $\overleftarrow{\gamma_1} = \{z, c, a, s\}$, $\overleftarrow{\gamma_2} = \{z, d, b, s\}$, $\overleftarrow{\gamma_3} = \{z, d, e, a, s\}$, $\overleftarrow{\gamma_4} = \{z, c, e, b, s\}$, $\overleftarrow{\gamma_5} = \{z, c, e, a, s\}$ *and* $\overleftarrow{\gamma_6} = \{z, d, e, b, s\}$. *We forget the direction of paths and reformulate it by adding the capacities of paths between the terminals. Set of paths in abstract auxiliary network is* $\tilde{\Gamma} = \{\tilde{\gamma}_1, \tilde{\gamma}_2, \tilde{\gamma}_3, \tilde{\gamma}_4, \tilde{\gamma}_5, \tilde{\gamma}_6\}$ *(cf. Figure 1).*

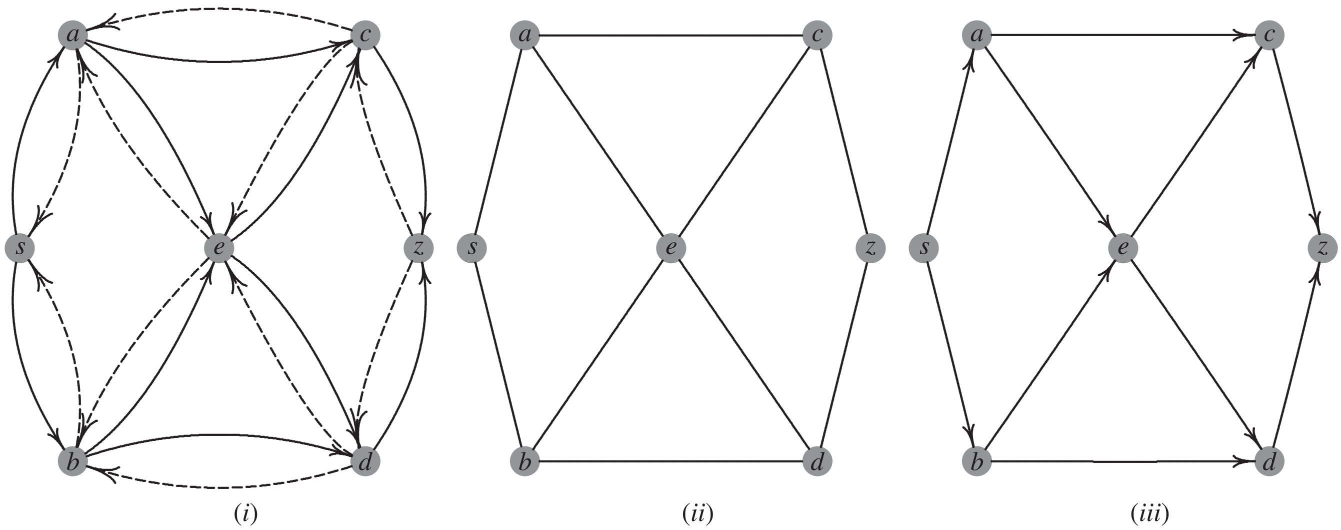


Figure 1. Abstract evacuation network, auxiliary network and network after contraflow reconfiguration, respectively

3.1 Maximum Abstract Contraflow

We define the maximum abstract contraflow (MACF) by integrating the contraflow model introduced in (Arulselvan, 2009; Rebennack *et al.*, 2010) and the MAF problem solved in (McCormick, 1996). With simple extension, we also give an efficient algorithm for solving Problem 1.

Problem 1. *Given an abstract network $N = (E, \Gamma, b, s, z)$, the MACF problem is to find the MAF with path reversals allowed initially.*

Theorem 1. *Let $\tilde{b} \in Z^+$ be the capacities and $\omega_{\tilde{\gamma}}$ be the supermodular weights on paths $\tilde{\gamma} \in \tilde{\Gamma}$. Then the weighted abstract flow and weighted abstract cut problems in auxiliary abstract network (AAN) have totally dual integer optimum solutions.*

To get solution for MACF, we combine the concepts of path reversal criterion presented in (Pyakurel *et al.*, 2017) and MAF algorithm provided in (McCormick, 1996).

Lemma 1. *The MACF does not decrease the flow value after contraflow configuration.*

Proof. By definition, $C \subseteq E$ is the collection of disconnecting and saturated elements such that every path connects source and sink, and contains only one element from $\tilde{\gamma}$. By construction, $\tilde{b}(\tilde{C}) = \sum_{e \in \tilde{C}} \tilde{b}_e \geq \sum_{e \in C} b_e$, where $\tilde{C} \subseteq \tilde{E}$ represents the cut in auxiliary network. Following Theorem 1, we have

$$val_{max}(\tilde{f}) = max \sum_{\tilde{\gamma} \in \tilde{\Gamma}} f(\tilde{\gamma}) = min\{\sum_{\tilde{e} \in \tilde{C}} \tilde{b}_{\tilde{e}} : \tilde{C} \subseteq \tilde{E}\} \geq min\{\sum_{e \in C} b_e : C \subseteq E\} = max \sum_{\gamma \in \Gamma} f(\gamma) = val_{max}(f).$$

Thus the claim follows. □

Lemma 2. *(Pyakurel et al., 2017) The abstract contraflow doubles the flow value after contraflow configuration whenever each element in a minimum abstract cut has symmetric capacity.*

Algorithm 1. *Maximum Abstract Contraflow Algorithm*

1. *Given path reversible abstract network $N = (G, b, s, z)$, where $G = (E, \Gamma)$.*
2. *Construct the AAN, $\tilde{N} = (\tilde{G}, \tilde{b}, \tilde{s}, \tilde{z})$ with new capacity $\tilde{b}(\tilde{\gamma}) = b(\overrightarrow{\gamma}) + b(\overleftarrow{\gamma})$.*
3. *Solve the maximum abstract network flow problem in $\tilde{N} = (\tilde{G}, \tilde{b}_e, \tilde{s}, \tilde{z})$ using (McCormick, 1996) as follows:*
   (a) *Initialize $\tilde{f} = \tilde{f}_0$, if an initial solution is given, otherwise initialize as the zero flow.*
   (b) *While $\tilde{f}$ is not optimal:*
      i. *Compute an augmenting structure. If no such structure exists, return $\tilde{f}$.*
      ii. *Determine $\delta \in \tilde{N}$ so that all paths in augmenting structure can be augmented by δ.*
      iii. *For each path $\tilde{\gamma}^+$ in augmenting structure, set $\tilde{f}_{\tilde{\gamma}^+} = \tilde{f}_{\tilde{\gamma}^+} + \delta$.*
      iv. *For each path $\tilde{\gamma}^-$ in augmenting structure, set $\tilde{f}_{\tilde{\gamma}^-} = \tilde{f}_{\tilde{\gamma}^-} - \delta$.*
4. *A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $b(\overrightarrow{\gamma})$ or there is a non-negative flow along the path $\overrightarrow{\gamma} \notin \Gamma$.*

Theorem 2. *Algorithm 1 solves Problem 1 optimally.*

Proof. The Steps 2 and 3 are feasible by definition. Step 4 is well defined; i.e. not both paths $\overrightarrow{\gamma}$ and $\overleftarrow{\gamma}$ have to be switched at a time, this is ensured by the solution of the abstract flow in auxiliary network, (Hoffman, 1974). Switching property cancels cycle flows so that there is flow along $\overrightarrow{\gamma}$ or $\overleftarrow{\gamma}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with path reversals in N.

Weights are supermodular and abstract path system satisfies switching property so that the MAF is totally dual integral with minimum cut, (Hoffman, 1974). By equivalence of optimization and separation problems of (Grötschel *et al.*, 1988), the oracle solves the separation problem for weighted abstract flow problem. The abstract maximum flow algorithm maintains a candidate set for an abstract minimum cut as solution of dual problem. The algorithm then calls oracle to verify the dual feasible solution, infact this is the case, the primal solution is a MAF, (McCormick, 1996). Otherwise, the oracle returns a violating path. The returned violated paths are then combined to an augmenting structure which allows to improve the flow value. In fact, any optimal solution to the maximum flow problem with path reversals on N is also a feasible solution to the maximum flow problem on $\tilde{N}$. As the amount of flow sent from s to z in Steps 3 is not changed in Step 4, the resulting flow is an optimal solution. □

Corollary 1. *(Pyakurel et al., 2017) Algorithm 1 computes MACF solution in polynomial time.*

Proof. The direction of paths can be reversed using Step 4 in linear time. Construction of AAN takes linear time. Thus, the complexity of Algorithm 1 depends on the complexity of Step 3 that takes $O(|\tilde{E}|logB)$ time in $\tilde{N}$, where B is the maximum capacity of any path $\tilde{\gamma} \in \tilde{\Gamma}$. □

Example 3. *Consider the abstract network of Figure 2 (i), where 4 units flow through $(s;b;z)$, 2 units flow through $(s;a;z)$ and 1 unit flow through $(s;b;a;z)$ can be send from source to sink. Here, the MAF is 7 in the network. But it is 13 for the network of Figure 2 (iii) after contraflow configuration with following path flows: 6 units through $(s;b;z)$, 6 units through $(s;a;z)$ and 1 units through $(s;b;a;z)$.*

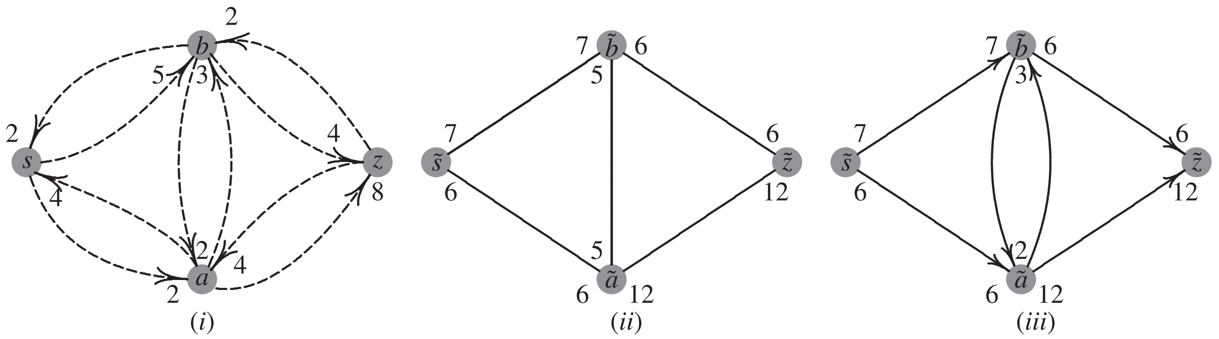


Figure 2. Abstract evacuation network, auxiliary network and network after contraflow reconfiguration, respectively

3.2 Maximum Dynamic Abstract Contraflow

The MDAF problem has been introduced and solved in (Kappmeier *et al.*, 2014). The maximum dynamic contraflow problem with arc reversal capability and its strongly polynomial time solution procedure are presented by (Rebennack *et al.*, 2010). We introduce and solve the maximum dynamic abstract contraflow (MDACF) with path reversal capability (cf. Problem 2).

Problem 2. *Given an abstract network $N = (E, \Gamma, b, \tau, s, z, T)$, the MDACF problem is to find the MDAF with path reversal capability at time zero.*

The abstract path system is allowed to be asymmetric with respect to the path capacities but the transit times are symmetric. In auxiliary network only capacities of the paths change but the transit times remain the same. As the abstract cut C_{dyn} contains an element of every temporally repeated paths, the capacity constraints are satisfied at each point of time. Thus for a subset $C_{dyn} \subseteq E_T$, the set $\gamma_t \cap C_{dyn}$ is nonempty to each $\gamma_t \in \Gamma_T$, (Kappmeier *et al.*, 2014). This implies that $\sum_{\gamma_t \in \Gamma_T} f_{dyn}(\gamma_t) \leq \sum_{(e,\,\theta) \in C_{dyn}} b_e$. The number of paths created by applying the time expansion is linear in T and thus exponential in the size of input. We combine the concept of path reversal capability of (Pyakurel *et al.*, 2017) and solution method of MDAF problem of (Kappmeier *et al.*, 2014). Algorithm 2 works in ATEN of an AAN that consists of a (static) abstract network. For each interval, a copy of the element set $\tilde{E}$, the element set $\tilde{E}_T := \tilde{E} \times \mathbf{T}$ will be constructed in ATEN is constructed by $\tilde{E}_T = \{(\tilde{e}, \theta) | \tilde{e} \in \tilde{E}, \theta \in \{1, 2, \cdots, T\}\}$.

Example 4. *The ATEN of an AAN can destroy the switching property. Let $(\tilde{E}, \tilde{\Gamma})$ be an auxiliary network with $\tilde{\Gamma} = \{\gamma^1, \gamma^2, \gamma^3, \gamma^4\}$ and $\tilde{E} = \{s, a, b, z\}$, where $\gamma^1 = (s;\ a;\ b;\ z)$, $\gamma^2 = (s;\ b;\ a;\ z)$, $\gamma^3 = (s;\ a;\ z)$, $\gamma^4 = (s;\ b;\ z)$ together with their reversals. The set $\tilde{\Gamma}$ satisfies the switching and order properties. For $T = 4$, we have $\Gamma_T = \{\gamma_0^1, \gamma_2^1, \gamma_0^2, \gamma_1^2, \gamma_0^3, \gamma_1^3, \gamma_2^3, \gamma_0^4, \gamma_1^4, \gamma_2^4\}$ with $\gamma_0^1 = \{(s, 0), (a, 1,), (b, 2), (z, 3)\}$ and $\gamma_1^2 = \{(s, 1), (b, 2), (a, 3), (z, 4)\}$. But time expansion of given network does not contain $\gamma_0^1 \times_{(b,2)} \gamma_1^2$ (cf. Figure 3).*

As in the abstract dynamic network, we can construct time-expanded abstract network $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ of the auxiliary abstract dynamic network $(\tilde{E}, \tilde{\Gamma})$ with time horizon T. In this case, internal waiting does not make difference in optimality, (Kappmeier *et al.*, 2014).

Lemma 3. *The MDACF in $N = (G, b, \tau, s, z, T)$ is not more than the optimal flow in the MACF problem for the corresponding time expanded graph.*

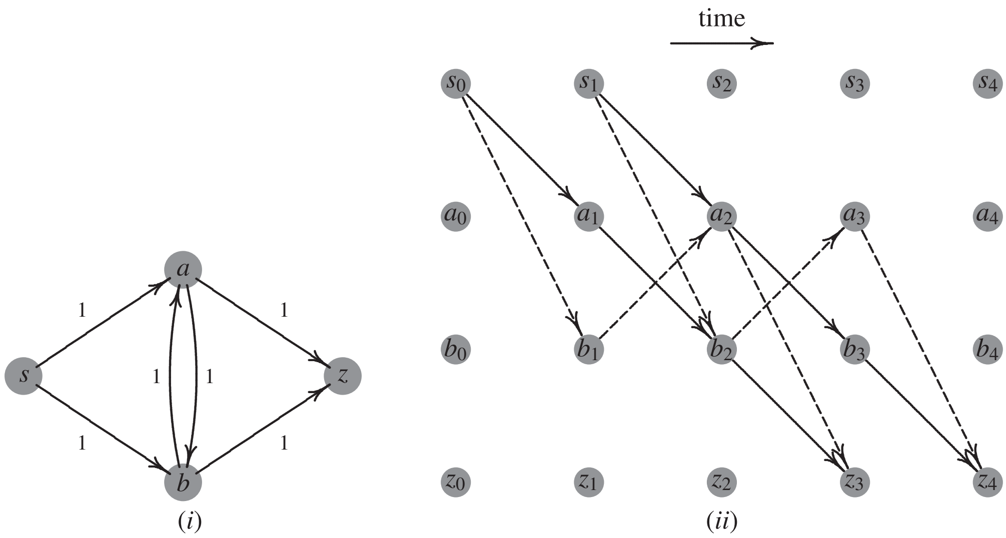


Figure 3. Dynamic abstract network and time expansions of the abstract network, respectively

Proof. The maximum amount of flow in the single source and single sink dynamic contraflow problem is less than the optimal flow in the maximum contraflow problem for the corresponding time expanded graph, (Rebennack *et al.*, 2010). The ATEN can be constructed by allowing intermediate waiting that does not make difference in optimality of MDAF, (Kappmeier *et al.*, 2014). Then we can conclude that every feasible flow to the MDACF problem has an equivalent feasible flow to the maximum contraflow problem of the time expanded graph. Hence, the claim follows. □

Lemma 4. *The dynamic abstract contraflow does not decrease the flow value after contraflow.*

Proof. By definition of minimum dynamic cut, $C_{dyn} \subseteq E_T$ is the collection of disconnecting and saturated elements with each path connecting source and sink in ATEN. The element set $\tilde{E}$ is copied for each point of time to construct the element set $\tilde{E}_T := \tilde{E} \times T$ in ATEN. The network $(\tilde{E}, \tilde{\Gamma})$ can destroy switching property. To construct abstract network, waiting is allowed in intermediate nodes but there is not any difference in optimality, (Kappmeier *et al.*, 2014). Every path passes through the cut set exactly onces on every time interval, i.e., $\tilde{\Gamma}^{\sigma} \cap \tilde{C}_{dyn} \neq \emptyset$, (Kappmeier *et al.*, 2014). The $\tilde{C}_{dyn}$ contains only one element from $\tilde{\gamma}$. The minimum cut capacity becomes $\tilde{b}(C^*_{dyn}) = \sum_{(\tilde{e},\theta)\in C^*} \tilde{b}_e\}$, where $\tilde{C}^*_{dyn}$ is the minimum cut. If each element $(e, \theta) \in C^*_{dyn}$ has capacity in both directions, the contraflow reconfiguration of the abstract network increases the capacity of cut but the capacity of minimum cut will not decrease even if there is not capacity in both directions. Then

$$\tilde{b}(\tilde{C}^*_{dyn}) = \sum_{(\tilde{e},\theta)\in\tilde{C}^*_{dyn}} \tilde{b}_{(\tilde{e},\theta)} \geq \sum_{(e,\theta)\in C^*_{dyn}} b_{(e,\theta)}.$$

Every MDAF is equal to the minimum dynamic abstract cut in auxiliary network, (Kappmeier *et al.*, 2014). Then,

$$\begin{aligned} val_{max}(\tilde{f}_{dyn}) &= max \sum_{\tilde{\gamma}_t\in\tilde{\Gamma}^{\sigma}_T} \tilde{f}(\tilde{\gamma}_t) = min\{\sum_{\tilde{e}\in\tilde{C}} \tilde{b}_{\tilde{e}} : \tilde{C}_{dyn} \subseteq \tilde{E}_T\} \\ &\geq min\{\sum_{e\in C} b_e : C_{dyn} \subseteq E_T\} = max \sum_{\gamma_t\in\Gamma^{\sigma}_T} f(\gamma_t) = val_{max}(f_{dyn}). \end{aligned}$$

The MDAF in AAN $\tilde{N} = (\tilde{G}, \tilde{b}_e, \tilde{\tau}, \tilde{s}, \tilde{z}, T)$ is a feasible dynamic abstract contraflow in original abstract network $N = (G, b_e, \tau, \tilde{s}, \tilde{z}, T)$ with path reversal capability. Hence the claim follows. □

Lemma 5. *The contraflow reconfiguration of an abstract contraflow network increases the flow value two times if each element in a minimum abstract dynamic cut has symmetric capacity.*

Algorithm 2. *Maximum Dynamic Abstract Contraflow Algorithm*

1. *Given abstract network $N = (G, b, \tau, \tilde{s}, \tilde{z}, T)$, where paths can be reverse without any cost.*
2. *Construct AAN, $\tilde{N} = (\tilde{G}, \tilde{b}_e, \tilde{\tau}, \tilde{s}, \tilde{z}, T)$ with new capacity and transit time functions $\tilde{b}(\tilde{\gamma}) = b(\overrightarrow{\gamma}) + b(\overleftarrow{\gamma})$ and $\tilde{\tau}(\tilde{\gamma}) = \sum_{\tilde{e}\in\tilde{\gamma}} \tilde{\tau}(\tilde{e})$.*
3. *Generate a temporally repeated dynamic flows $\tilde{G}$ with capacity $\tilde{b}(\tilde{\gamma})$ and transit time $\tilde{\tau}(\tilde{\gamma})$, (Kappmeier et al., 2014).*
4. *A path $\overleftarrow{\gamma} \in \Gamma$ is reversed if and only if the flow along $\overrightarrow{\gamma} \in \Gamma$ is greater than $b(\overrightarrow{\gamma})$ or there is a non negative flow along path $\overrightarrow{\gamma} \notin \Gamma$.*

Theorem 3. *Algorithm 2 solves Problem 2 optimally.*

Proof. The Steps 2 and 3 are feasible. Constructed flows from Step 3 are temporally repeated, there is only a flow in one direction of two elements at the same time as well as at different time periods, this ensures that the flow is less than the reversed capacities on all the paths at all time units and also ensures the feasibility of Step 4. Since every feasible flow of the DAF problem in $\tilde{N}$ is feasible to the MDACF problem in N, the MDACF in N is not more than the MDAF in $\tilde{N}$. The MACF in ATEN is not less than the MDACF in N, Lemma 3. Hence, the optimal dynamic contraflow in N is not greater than the MACF in N_T. The MACF problem in N_T is equivalent to the maximum flow problem in $\tilde{N}_T$, Theorem 2. Thus, the MACF in N_T is equal to the maximum abstract flow (MAF) in $\tilde{N}_T$. The temporally repeated abstract flow f^*_{dyn} is a MDAF, and C_{dyn} is a minimum abstract cut over time whose capacity is equal to the flow value, (Kappmeier *et al.*, 2014). Thus the maximum flow can be obtained by a temporally repeating a path flow of a static graph $\tilde{N}$. Hence the MAF in $\tilde{N}_T$ is equal to the MDAF in $\tilde{N}$. □

Corollary 2. *Algorithm 2 computes the MDACF in polynomial time.*

Proof. Steps 2 and Step 4 can be completed in linear time, so that the complexity is dominated by Step 3 that computes the MDAF problem in AAN. Step 3 can be computed in polynomial time, (Kappmeier *et al.*, 2014). □

Example 5. *Consider the symmetric transit time $\tau_e = 1$ for each $e \in E$ in Figure 2 (i). Using the algorithm, 6 units flow at time 4 through the paths $(s; a; z)$ and $(s; b; z)$, 7 units flow at time 5 through the paths $(s; a; z)$, $(s; b; z)$ and $(s; b; a; z)$ can be send to the sink. By the same idea, dynamic flows can be calculated in the network after contraflow configuration. The MDAF at time $T = 5$ are 13 and 25 before and after contraflow configuration, respectively, showing the significant increment with contraflow.*

3.3 Lexicographically Maximum Abstract Contraflow

If we have a given rank on the terminals with priorities, the flow is compared by its value in their rank ordering, referred to as lexicographically maximum flow. An existence and a polynomial solution of LMAF problem have been presented in (Kappmeier, 2015). In his model, the order of terminals has to fulfill compatible property if more than one terminal node is contained in a path.

Problem 3. *Let $N = (G, b, S, D)$ be an abstract network, where $G = (E, \Gamma)$ contains compatible sets of sources S and sinks D. The lexicographically maximum abstract contraflow (LMACF) problem is to find a LMAF where paths can be reversed without any cost.*

Based on the LMACF algorithm, (Kappmeier, 2015) and the lexicographically maximum contraflow algorithm (Pyakurel & Dhamala, 2015), we propose LMACF Algorithm 3, which solves the Problem 3 in polynomial time.

Algorithm 3. *Lexicographically Maximum Abstract Contraflow Algorithm*

1. *Given abstract network $N = (G, b, S, D, \omega \equiv 1)$ with a compatible sequence of sources $s_1, s_2, \cdots, s_k$ or sinks $z_1, z_2, \cdots, z_k$ in S and D, respectively.*
2. *Construct AAN, $\tilde{N} = (\tilde{G}, \tilde{b}_e, \tilde{S}, \tilde{Z}, \omega \equiv 1)$ with capacity $\tilde{b}(\tilde{\gamma}) = b(\overrightarrow{\gamma}) + b(\overleftarrow{\gamma})$.*

3. *Find solution in auxiliary network using Abstract Lexicographically Maximum Flow Algorithm, (Kappmeier, 2015):*
   (a) *Set* $i = 0$ *and initialize* $\tilde{f}^0 = 0$ *as the zero flow on all paths.*
   (b) *Set* $i = i + 1$ *and define the abstract networks* $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$, *(or* $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{z}})$ *w. r. t. the sinks).*
   (c) *Compute a flow* $\tilde{f}^i$ *using Step 3 of Algorithm 1 in* $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$ *starting with solution* $\tilde{f}^{i-1}$.
   (d) *If* $i = k$ *return* $\tilde{f}^k$, *otherwise continue with Step 3b.*
4. *A path* $\overleftarrow{\gamma} \in \Gamma$ *is reversed if and only if the flow along* $\overrightarrow{\gamma} \in \Gamma$ *is greater than* $b(\overrightarrow{\gamma})$ *or there is a non negative flow along path* $\overrightarrow{\gamma} \notin \Gamma$.

Lemma 6. *Let* $\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_k$ *be a compatible sequence of sources and let* $\tilde{f}^i$ *be a MAF in the auxiliary network* $(\tilde{A}, \tilde{\Gamma}^i_{\tilde{s}})$. *If LMAF algorithm is executed in* $(\tilde{A}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$ *with initial flow* $\tilde{f}^i$ *as input, the flow value* $|\tilde{f}|^+_{\tilde{s}_j}$ *does not decrease for* $j = 1, \cdots, i$ *during the execution. Moreover, the same is true for sinks.*

Theorem 4. *Algorithm 3 solves Problem 3 in* $N = (G, b, S, D, \omega \equiv 1)$ *optimally.*

Proof. The feasibility can be proved as in Theorem 2. The path capacity is increased by adding the capacity of paths and either direction of them is allowed in auxiliary network with unaltered priority ordering. Step 3 works for source and sink element sequences in auxiliary network. We prove the optimality by induction on i. The first iteration computes a MAF $\tilde{f}^1$ in $(\tilde{E}, \tilde{\Gamma}^1_{\tilde{s}})$ with single source. Let $\tilde{f}^i$ be a LMAF in $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$. By Lemma 6, flow $\tilde{f}^{i+1}$ does not reduce the inflow of source element $\tilde{s}_j$ for some $j \in \{1, 2, \cdots, i\}$ and the flow is maximum. Assume now that $\tilde{f}^{i+1}$ is not a LMAF in the abstract network $(\tilde{E}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$. Then there is a flow $\tilde{f}'$ that sends more flow out of the source $\tilde{s}_j$ for some $j \in \{1, 2, \cdots, i\}$. Define the restricted $\tilde{f}^r$ by setting $\tilde{f}^r = \tilde{f}'$ for each path $\gamma \in \Gamma^i$. The outflow of source element $\tilde{s}_j$ is the same for $\tilde{f}'$ and $\tilde{f}^r$, and $\tilde{f}^r$ is a feasible abstract flow in $(\tilde{E}, \tilde{\Gamma}^i_{\tilde{s}})$ that sends more flow out of $\tilde{s}_j$ than does $\tilde{f}^i$ which contradicts that $\tilde{f}^i$ is a lexicographically maximum. Hence, $\tilde{f}^{i+1}$ is a LMAF in $(\tilde{E}, \tilde{\Gamma}^{i+1}_{\tilde{s}})$. Any optimal solution to the LMAF problem with path reversal in graph (E, Γ) is also a feasible solution to the maximum flow problem in $\tilde{N}$. As the amount of flow sent from $\tilde{S}$ to $\tilde{D}$ in Step 3 is not changed in Step 4, the resulting flow is optimal for the Problem 3. □

Corollary 3. *Algorithm 3 solves the LMACF problem in polynomial time complexity.*

Proof. The construction of auxiliary network and Step 4 can be completed in linear time. As $N = (G, b_e, S, D, \omega \equiv 1)$ is an abstract network with a compatible sequence of sources, the complexity depends upon the running time of Step 3. Moreover, a separation oracle is given in the problem. The LMAF can be computed in polynomial time which is dominated by $|\tilde{E}|$, (Kappmeier, 2015). □

3.4 Earliest Arrival Abstract Contraflow

The existence of earliest arrival abstract flow (EAAF) described by (Kappmeier, 2015) generalizes the earliest arrival flow which maximizes the DAF at each possible time point.

Problem 4. *Let* $N = (G, b, \tau, s, z, T)$ *be an abstract dynamic network, where* $G = (E, \Gamma)$. *The earliest arrival abstract contraflow (EAACF) problem is to find the EAAF with path reversal capability at time zero.*

Based on the earlier results of earliest arrival contraflow problem with arc reversals in (Pyakurel & Dhamala, 2015) and EAAF in (Kappmeier, 2015), we introduce EAACF problem and propose Algorithm 4 to solve it. (Minieka, 1973) proved that using the successive shortest path algorithm in the original network and sending flow along the generalized temporally repeated paths leads to an earliest arrival flow. This approach does not require waiting in intermediate nodes and sends flow only along temporal copies of the original paths. A similar algorithm to the successive shortest path algorithm is in (Martens and McCormick, 2008). Their algorithm computes a maximum weighted abstract flow by using augmenting structure of decreasing total reward, where the shortest path has the most reward.

Algorithm 4. *Earliest Arrival Abstract Contraflow Algorithm*

1. *Given abstract network* $N = (G, b, \tau, s, z, T)$ *where paths can be reverse without any cost.*
2. *Construct AAN,* $\tilde{N} = (\tilde{G}, \tilde{b}_e, \tilde{\tau}_e, \tilde{s}, \tilde{z})$ *with new capacity and transit functions* $\tilde{b}(\tilde{\gamma}) = b(\overrightarrow{\gamma}) + b(\overleftarrow{\gamma})$ *and* $\tilde{\tau}(\tilde{\gamma}) = \sum_{\tilde{e} \in \tilde{\gamma}} \tilde{\tau}(\tilde{e})$, *respectively.*

3. *Solve the problem in the auxiliary network using Step 3 of Algorithm 3 in corresponding* $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ *of* $\tilde{N}$.

4. *A path* $\overleftarrow{\gamma} \in \Gamma$ *is reversed if and only if the flow along* $\overrightarrow{\gamma} \in \Gamma$ *is greater than* $b(\overrightarrow{\gamma})$ *or there is a non negative flow along path* $\overrightarrow{\gamma} \notin \Gamma$.

Theorem 5. *Algorithm 4 solves Problem 4 optimally.*

Proof. The feasibility can be ensured as in Theorem 2. Let $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ be the corresponding increasing ATEN of $\tilde{N}$ with intermediate waiting, (Kappmeier *et al.*, 2014), where T be a significantly large time horizon. Step 3 is executed in $(\tilde{E}_T, \tilde{\Gamma}_T^{\sigma})$ that avoids reduction of flow sent in earlier time steps by removing backward paths in the residual network, Lemma 6. For an abstract network $(\tilde{E}, \tilde{\Gamma}, T)$, there exists an EAAF flow $\tilde{f}$ in $(\tilde{E}, \tilde{\Gamma}, T)$, (Kappmeier, 2015). The resulting flow $\tilde{f}$ from Step 3 signifies that the obtained flow is optimal in $(\tilde{E}, \tilde{\Gamma}, T)$. Any optimal solution to the earliest arrival flow problem with path reversal in network N is also a feasible solution to the EAAF problem in the auxiliary network $\tilde{N}$. As the amount of flow sent from s to z in Step 3 is not changed in Step 4, the resulting flow is an optimal solution to the EAACF problem in N. □

Example 6. *Consider the network of Example 5. In this example, MDACF satisfies the earliest arrival flow property. Here, the EAAF values before contraflow configuration at time* $T = 4$ *and* 5 *are* 6 *and* 13*, respectively. Also, the EAACF values at these times are* 12 *and* 25*, respectively.*

4. Conclusions

In this paper, we studied both abstract flow and contraflow models from litrature. Integrating these models, we introduced abstract contraflow approach with discrete time settings for the first time. Through these models we came to know that the switching property is the most essential force behind abstract flow. In abstract flow model, some structural results of classical network are also valid such as, the MAF and minimum abstract cut are strong dual to each other. We proposed efficient algorithms for maximum dynamic and EAACF problems in two-terminal abstract networks. We also proposed polynomial time algorithm for lexicographically maximum abstract contraflow. Our results increase the flow values by reducing crossing conflicts with arc reversals toward the safe destinations in evacuation planning.

Acknowledgements

The first author thanks the University Grants Commission, Nepal for its support. The authors acknowledge the supports of DAAD and Alexander von Humboldt Foundation.

References

Altay, N., & Green III, W. G. (2006). Or/ms research in disaster operations management. *European Journal of Operational Research, 17*, 475-493.

Arulselvan, A. (2009). Network model for disaster management, PhD thesis, University of Florida.

Dhamala, T. N. (2015). A survey models algorithms for discrete evacuation planning network problems. *Journal of Industrial and Management Optimization, 11(1)*, 265-289.

Dhamala, T. N., & Pyakurel, U. (2013). Earliest arrival contraflow problem on series parallel graph. *International Journal of Operations Research, 10(1)*, 1-13.

Dhungana, R. C., Pyakurel, U., Khadka, S. R., & Dhamala, T. N. (2015). Universally maximum contraflow on for evacuation planning. *International Journal of Operations Research Nepal, 4(1)*, 67-78.

Fleischer, L., & Skutella, M. (2003). The quickest multicommodity flow problem. *Lecture Notes in Computer Science*, 36-53.

Ford, L. R., & Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Mathematical Society, 8*, 399-404.

Ford, L. R., & Fulkerson, D. R. (1958). Constructing maximal dynamic flows from static network. *Operations Research, 6*, 416-433.

Grötschel, M., Lovasz, L., & Schrijver, A. (1988). Geometric algorithms and combinatorial optimization. Vol. 2, Springer, Berlin, Heidelberg.

Hamacher, H. W., & Tjandra, S. A. (2001). Mathematical modeling of evacuation problems: a state of the art, in pedestrian and evacuation dynamics, (eds. M. Schreckenberger and S.D. Sharma), Springer.

Hoffman, A. (1974). A generalization of max flow-min cut. *Mathematical Programming, 6(1)*, 352-359.

Hoppe, B., & Tardos, E. (1994). Polynomial time algorithm for some evacuation problems. *Annual ACM-SIAM symposium on Discrete Algorithms, 5*, 433-441.

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

Hua, J., Ren, G., Cheng, Y., & Ran, B. (2014). An integrated contraflow strategy for multi modal evacuation. *Hindawi Publishing Corporation Mathematical Problems in Engineering*. Article ID 159473.

Jarvis, J. J., & Ratliff, H. D. (1982). Some equivalent objectives for dynamic network flow problems. *Management Science, 28(1)*, 106-109.

Kappmeier, J.-P. W. (2015). Generalizations of flows over time with application in evacuation optization, PhD thesis, Technical University, Berlin, Germany.

Kappmeier, J.-P. W., Matuschke, J., & Peis, B. (2014). Abstract flows over time: a first step towards solving dynamic packing problems. *Theoretical Computer Science, Algorithms and Computation, 544*, 74-83.

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

Martens, M., & McCormick, S. T. (2008). A polynomial algorithm for weighted abstract flow. *Integer Programming and Combinatorial Optimization*, 97-111.

McCormick, S. T. (1996). A polynomial algorithm for abstract maximum flow. *In: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, 490-497.

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

Minieka, E. (1973). Maximal lexicographic and dynamic network flows. *Operations Research, 21*, 517-527.

Moriarty, K., Ni, D., & Collura, J. (2007). Modeling traffic flow under emergency evacuation situations: current practice and future directions. *In: 86th Transportation Research Board Annual Meeting, Washington, D.C., 07-0745*.

Pel, A. J., Bliemer, M. C. J., & Hoogendoorn, S. P. (2012). A review on travel behavior modeling in dynamic traffic simulation models for evacuations. *Transportation, 39*, 97-123.

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

Pyakurel, U. (2016). Evacuation planning problem with contraflow approach, PhD thesis, Tribhuvan University, Kathmandu, Nepal.

Pyakurel, U., & Dhamala, T. N. (2016). Continuous time dynamic contraflow models and algorithms. *Hindawi Publishing Corporation, Advances in Operations Research*. Article ID 7902460.

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

Pyakurel, U., & Dhamala, T. N. (2017b). Evacuation planning by earliest arrival contraflow. *Journal of Industrial and Management Optimization, 13*(1), 489-503.

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

Pyakurel, U., Hamacher, H. W., & Dhamala, T. N. (2014). Generalized maximal dynamic contraflow on lossy network. *International Journal of Operations Research Nepal, 3(1)*, 27-44.

Pyakurel, U., Nath, H. N., & Dhamala, T. N. (2018). Effcient contraflow algorithms for quickest evacuation planning. *Science China Mathematics*. Doi 10.1007/s11425-017-9264-3.

Rebennack, S., Arulselvan, A., Elefteriadou, L., & Pardolas, P. M. (2010). Complexity analysis of maximum flow problem with arc reversal. *Journal of Combinatorial Optimization, 29*, 200-216.

Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., & Seyfried, A. (2009). Evacuation dynamics: Empirical results, modeling and applications. *In Encyclopedia of Complexity and Systems Science*, 3142-3176.

Wang, J., Wang, H., Zhang, W., Ip, W. H., & Furuta, K. (2013). Evacuation planning based on the contraflow technique with consideration of evacuation priorities and traffic setup time. *IEEE Transactions on Intelligent Transportation Systems, 14*, 480-485.

Xie, C., Lin, D. Y., & Waller, S. (2010). A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. *Transportation Research Part E, 46*, 295-316.

Xie, C., & Turnquist, M. A. (2011). Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach. *Transportation Research Part C, 19*, 40-63.

Zhao, X., Feng, Z.-Y., Li, Y., & Bernard, A. (2016). Evacuation network optimization model with lane-based reversal and routing. *Hindawi Publishing Corporation, Advances in Operations Research*. Article ID 1273508.

Copyrights

The Nepali Math. Sc. Report
Vol. 35, No.1 and 2, 2018

ABSTRACT EARLIEST ARRIVAL TRANSSHIPMENT WITH NETWORK RECONFIGURATION

RAM CHANDRA DHUNGANA[1] AND TANKA NATH DHAMALA[2]

[1,2] *Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal*
[1] *dhungana.ramchandra@gmail.com* and [2] *amb.dhamala@daadindia.org*

Abstract: The abstract flow model is the generalization of network flow model which deals with the flow paths (routes) satisfying the switching property. Contraflow model increases the flow value by reversing the required arc directions from the sources to the sinks. In this paper, we integrate the concepts of abstract flow and contraflow to introduce abstract earliest arrival transshipment contraflow model in multi-terminal abstract network. The abstract contraflow on multi-terminal dynamic network is NP-Complete. We present an efficient approximation algorithm to solve the problem. This approach satisfies the demand of sinks by sending optimal flow at every possible time point and seeks to eliminate the crossing conflicts.

Key Words: Abstract flow, contraflow, earliest arrival flow, transshipment, evacuation network
MSC(2010) Primary: 90B10, 90C27, 68Q25; Secondary: 90B06, 90B20.

1. Introduction

The first network flow model was developed in [6]. After the development of maximal static flow and maximum dynamic flow models, Different researchers have studied several network flow problems for evacuation planning. The maximum dynamic flow problem to shift maximal amount in a given time, the earliest arrival flow problem to maximize the number of evacuees in every possible time, the quickest flow problem for allocating the evacuees to a safer zone in minimum time, the lexicographically maximum dynamic flow problem to send maximum number of evacuees in given priority order within the given time period, are few examples. According to Hoffman [7], the abstract flow generalizes the concept of paths by replacing the underlying network configuration. This tactic makes the use of switching property that eliminates the crossing at intersections. The maximum weighted abstract flow model has been developed by Hoffman [7] and solved by Martens and McCormick [12]. Kappmeier *et al.* [10] have investigated the maximum dynamic abstract flow problem and its solution procedure. The existence of the lexicographically maximum abstract flow problem for prioritized terminals has been shown in Kappmeier [9]. The earliest arrival abstract flow introduced and solved in [9] maximize the dynamic abstract flow from the source to the sink at every possible time. The abstract earliest arrival transshipment has been approximated for the fixed demands and supplies in [9]. Zhao *et al.* [23] deliberate the important lane-based routing strategy for reducing the interruptions that reduce (or eliminate) crossing and merging conflicts at nodes. Selecting and turning options at nodes are limited to expand traffic flow away from a unsafe area in a lane-based routing plan.

The contraflow approach introduced in [11] is another emerging and widely accepted model for evacuation planning. It increases the outbound road capacities by reversing the direction of roads towards the safe destinations. They give programming formulation and presented a greedy heuristic and a bottleneck relief heuristic for large scale evacuations that find high quality solution. They proved that problem of reducing

Received July 17, 2018 .

the egress time on evacuation network is $\mathcal{NP}$-hard. The first heuristic solves a minimum cost problem in the time-expanded network in the given time period to record the total number of flow units that pass through each arc during the evacuation time and flips the direction of each arc in favour of the direction of larger flows. The use of contraflow approach is not only bounded in evacuation planning, it also can be used in traffic arrangements and to reduce road accidents. Several mathematical models, heuristics, optimization and simulation techniques taking into account of macroscopic and microscopic behavioural characteristics deal with contraflow for this transportation network, however a suitable contraflow solution is still lacking due to very high computational costs.

The maximum dynamic contraflow problem has been introduced in [20] and solved with polynomial time algorithm for two terminal general network. The maximum dynamic contraflow problem on multi-terminal network is $\mathcal{NP}$-complete, [20]. Authors in [3, 17, 16] presented the earliest arrival, quickest transshipment and lex-maximum dynamic contraflow problems in both discrete and continuous time settings. The earliest arrival contraflow on series-parallel network can be solved in strongly polynomial time but the procedures on general networks is pseudo-polynomial. The quickest transshipment contraflow problem has been solved in polynomial time. For the given priority ordering, the lex-maximum dynamic contraflow problem is solved in polynomial time for multi-terminal network. For the fixed supplies and demands, the earliest arrival transshipment contraflow has been introduced in [18, 16] in discrete and continuous times. They solved the problems in multi-source and single-sink network as well as single-source and multi-sink network. In both cases, their algorithms have polynomial time complexity. Moreover, they presented approximation algorithms to compute the approximate earliest arrival transshipment contraflow for multi-terminal networks, [18, 19]. Moreover, authors in [19] introduced the abstract contraflow approach with path reversal capability. They presented a polynomial time algorithm to solve the abstract maximum dynamic contraflow in continuous time setting. Authors in [15] investigated quickest contraflow problems with constant and load dependent transit times. Various network flow models have been extended to the partial contraflow models in [14]. They have also presented solution procedures for different partial contraflow problems. For more details, we refer to [2].

The lane based contraflow and crossing elimination strategies at intersections is solved by the authors [22]. As introduced in [23], a network optimization model is to integrate these problems. The study [21], formulates the bi-level lane-based network optimization and simulation model, where the upper level optimizes the network evacuation performance subject to the contraflow and crossing-elimination constraints, and the lower level simulates dynamic evacuation flows. The multi-model integrated contraflow contains non-contraflow to shorten the strategy setup time, full-lane contraflow to minimize the evacuation network capacity and bus contraflow to realize the transit cycle operation. A more realistic contraflow problem with evacuation priorities and the setup time has been considered, (see in [2]).

The notations and prior works in network flows are presented in Section 2. The abstract flow and abstract contraflow models with their solution algorithms have been discussed in Section 3. The earliest arrival abstract contraflow model on multi-terminal abstract network has been introduced in Section 4. Also, the 2- value approximation algorithm for the problem is proposed . Section 5 concludes the paper.

2. Denotations and Basic Models

2.1. **Abstract Network Flows.** Consider a multi-terminal evacuation network $N = (E, \Gamma, b, \tau, S, D, T)$ where E and Γ represent the sets of elements and paths, respectively. Let $b : E \longrightarrow Z^+$ be the capacity function and $\tau : E \longrightarrow Z^+$ be the transit time function. The given non-negative time horizon T is symbolized by $T = \{0, 1, \cdots, T\}$ in discrete time setting, whereas it is denoted by $T = \{[0, 1), \cdots, [T, T + 1)\}$ in continuous time setting. For every path $\gamma \in \Gamma$ there is a linear order $<_\gamma$ of elements and the set of such paths Γ satisfies the switching property in abstract network setting. A switching property requires that for each $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, there exist paths $\gamma_1 \times_e \gamma_2 \subseteq \{a \in \gamma_1 : a \leqslant_{\gamma_1} e\} \cup \{a \in \gamma_2 : a \geqslant_{\gamma_2} e\}$ and $\gamma_2 \times_e \gamma_1 \subseteq \{a \in \gamma_2 : a \leqslant_{\gamma_2} e\} \cup \{a \in \gamma_1 : a \geqslant_{\gamma_1} e\}$. The before and after parts from e of the path γ excluding e are denoted by $(\gamma, e) = \{p \in \gamma : p <_\gamma e\}$ and $(e, \gamma) = \{p \in \gamma : p >_\gamma e\}$, respectively. Let

$f : \Gamma \longrightarrow R^+$ be the flow function. The generalized weighted maximum abstract flow problem formulated in [7] optimizes the value of Equation 2.1 satisfying Constraints 2.2 and 2.3.

$$\max \sum_{\gamma \in \Gamma} \omega(\gamma) f(\gamma) \tag{2.1}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} f(\gamma) \leq b_e, \ \forall\, e \in E \tag{2.2}$$

$$f(\gamma) \geq 0, \ \forall\, \gamma \in \Gamma \tag{2.3}$$

The weight function $\omega : \Gamma \longrightarrow R^+$ generalize the maximum abstract flow problem by specifying the reward per unit of flow sent along each path. The choice of the weighted function ω is restricted to supermodular functions, i.e., $\omega(\gamma_1 \times_e \gamma_2) + \omega(\gamma_2 \times_e \gamma_1) \geq \omega(\gamma_1) + \omega(\gamma_2)$ for every $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, as the general problem is $\mathcal{NP}$-hard. The generalized minimum weight abstract cut is the dual of the generalized maximum weighted abstract flow problem, [7], which can be formulated as:

$$\min \sum_{e \in E} b_e y(e) \tag{2.4}$$

$$\sum_{e \in \gamma} y(e) \geq \omega(\gamma), \ \forall\, \gamma \in \Gamma, \tag{2.5}$$

$$y(e) \geq 0, \ \forall\, e \in E \tag{2.6}$$

where a value $y(e)$ is assigned to every element $e \in E$ covering every path according to its weight.

The consideration of time factor transforms the abstract flow model into dynamic abstract flow model. The dynamic abstract flow problems can be transformed into static network by constructing the corresponding time expended networks. The time expanded network can violate the switching property. So, Kappmeier *et al.* [10] introduced the holdover of flow at intermediate nodes to construct an abstract time expanded network. Suppose $\sigma : \Gamma \to \{1, 2, \cdots, T\}$ be waiting times for every elements of Γ. Then the flow enters to $e \in \gamma$ at time $\sum_{q \in \gamma_{\to e}} (\sigma_q + \tau_q) + \tau_e$ as it travelling along γ waits σ_q time units before passing through e. The set $\gamma^\sigma = \{e^\kappa \in E^T | e \in \gamma, \sum_{q \in \gamma_{\to e}} (\sigma_q + \tau_q) + \tau_e = \kappa\}$ represents temporal paths with intermediate waiting and satisfy $q^\kappa <_{\gamma^\sigma} e^{\kappa'}$ if and only if $q <_\gamma e$. Such paths arrived within T is denoted by $\Gamma_T^\sigma = \{\gamma^\sigma | \gamma \in \Gamma, \sigma \in \{1, 2, \cdots, T\}^\Gamma, \sum_{e \in \gamma} \{\sigma_e + \tau_e\} \leq T\}$. The set (E_T, Γ_T^σ) represents the abstract time expanded network of (E, Γ).

Suppose $f_{dyn} : \Gamma_T^\sigma \to R_+$ be the dynamic abstract flow. The feasible dynamic abstract flow satisfies the capacity of every element at every point of time. The maximum dynamic abstract flow problem maximizes the total flow value respecting the given restrictions, [10].

$$\max \sum_{\gamma_t \in \Gamma_T^\sigma} f_{dyn}(\gamma_t) \tag{2.7}$$

$$\sum_{\gamma_t \in \Gamma_T^\sigma : (e,\theta) \in \gamma_t} f_{dyn}(\gamma_t) \leq b_e, \ \forall\, e \in E, \ \theta \in \mathbf{T} \tag{2.8}$$

$$f_{dyn}(\gamma_t) \geq 0, \ \forall\, \gamma_t \in \Gamma_T^\sigma \tag{2.9}$$

The dynamic cut is $C_{dyn} = \{(e, \theta) \in E_T : \alpha(e) \leq \theta < \alpha(e) + \tilde{y}(e)\}$, where $\tilde{y}$ is static weighted abstract dual integral optimal solution with weight $\omega(\gamma)$ and $\alpha(e) := \min_{\gamma \in \Gamma} \sum_{\gamma \in (\Gamma, e)} (\tau(\gamma) + \tilde{y}(\gamma))$.

Suppose $\gamma_1, \gamma_2 \in \Gamma$ are two paths with a common element $e \in \gamma_1 \cap \gamma_2$. The abstract network is terminal respecting, if all paths $\gamma \in \gamma_{1_{\to e}} \cup \gamma_{2_{e \to}}$ satisfy $\text{first}(\gamma) = \text{first}\,(\gamma_1)$ and $\text{last}\,(\gamma) = \text{last}(\gamma_2)$. Let $d^+ : E \to Q_+$ be supplies for source elements and let $d^- : E \to Q_+$ be demands for sink elements. An abstract flow satisfies given supplies and demands if $\sum_{\gamma \in \Gamma : e = \text{first}(\gamma)} f_\gamma = d_e^+$ and $\sum_{\gamma \in \Gamma : e = \text{ last}(\gamma)} f_\gamma = d_e^-$ holds for source and sink elements, respectively. It is possible that an element is a source and a sink both.

Let f be an abstract flow in (E_T, Γ_T^σ). For any $t \in \{1, 2, \cdots, T\}$, consider the set of temporal paths $\Gamma_t^\sigma = \{\gamma \in \Gamma | \sum_{e \in \gamma} \{\sigma_e + \tau_e\} \leq t\}$. Define the flow with value $|f_{dyn}|_t = \sum_{\gamma \in \Gamma^t} f_\gamma$ that arrives until time t. An earliest arrival abstract flow problem is to find a maximum flow such that it holds $|f_{dyn}|_t \geq |f_{dyn}^t|$ for all t, where $|f_{dyn}^t|$ is the value of a maximum dynamic abstract flow with time horizon t.

2.2. **Contraflow Reconfiguration.** The core concept behind the contraflow technique is to improve the outbound capacity by adopting the path reversals toward the sinks keeping the same travel time in the network. As a result the flow value is increased and egress time is degreased. Let $\Gamma = \{\overrightarrow{\gamma}, \overleftarrow{\gamma}\}$ be the set of all paths in contraflow abstract network N with capacities $b(\overrightarrow{\gamma}) = \min\{b_e : e \in \overrightarrow{\gamma}\}$ and $b(\overleftarrow{\gamma}) = \min\{b_e : e \in \overleftarrow{\gamma}\}$. We define the undirected auxiliary network $\tilde{N}$ by adding the capacities on the corresponding two-way paths and keeping the transit time (if any) fixed. The set of elements and paths are denoted by $\tilde{E}$ and $\tilde{\Gamma}$, where $\tilde{e} \in \tilde{E}$ and $\tilde{\gamma} \in \tilde{\Gamma}$. Then the capacity function is defined as $b(\tilde{\gamma}) = \min\{b_{\tilde{e}} : \tilde{e} \in \tilde{\gamma}\}$ while the travel times (if any) on paths remains the same. The auxiliary network resulting from path reversal holds order of elements for each $\tilde{\gamma} \in \tilde{\Gamma}$ and $\tilde{\Gamma}$ satisfies the switching property. Thus, the auxiliary network of the abstract evacuation network is also an abstract network.

3. Abstract Contraflow Problems

In this section, we discuss abstract flow, contraflow and abstract contraflow problems with their solution status from different literatures. The contraflow approach makes use of path reversals in abstract network at time zero without any switching costs.

Example 3.1. Let $E = \{s, a, b, z\}$ and $\Gamma = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overrightarrow{\gamma_3}, \overrightarrow{\gamma_4}, \overleftarrow{\gamma_1}, \overleftarrow{\gamma_2}, \overleftarrow{\gamma_3}, \overleftarrow{\gamma_4}\}$ with $\overrightarrow{\gamma_1} = \{s, a, z\}$; $\overrightarrow{\gamma_2} = \{s, a, b, z\}$; $\overrightarrow{\gamma_3} = \{s, b, z\}$; $\overrightarrow{\gamma_4} = \{s, b, a, z\}$; $\overleftarrow{\gamma_1} = \{z, a, s\}$; $\overleftarrow{\gamma_2} = \{z, b, a, d\}$; $\overleftarrow{\gamma_3} = \{z, b, s\}$ and $\overleftarrow{\gamma_4} = \{z, a, b, s\}$ and capacities $4, 2, 2, 2, 3, 3, 4$ and 2, respectively. We forget the direction of paths and reformulate it by adding the capacities of paths between the terminals. Set of paths in abstract auxiliary network is $\tilde{\Gamma} = \{\tilde{\gamma_1}, \tilde{\gamma_2}, \tilde{\gamma_3}, \tilde{\gamma_4}\}$ (cf. Figure 1). After contraflow reconfiguration, the path set becomes $\Gamma_R = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overrightarrow{\gamma_3}, \overrightarrow{\gamma_4}\}$ with improved capacities $7, 2, 6$ and 3 for $\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overrightarrow{\gamma_3}$, and $\overrightarrow{\gamma_4}$, respectively. Set of paths in abstract auxiliary network is $\tilde{\Gamma} = \{\tilde{\gamma_1}, \tilde{\gamma_2}, \tilde{\gamma_3}, \tilde{\gamma_4}, \tilde{\gamma_5}, \tilde{\gamma_6}\}$ (cf. Figure 1).

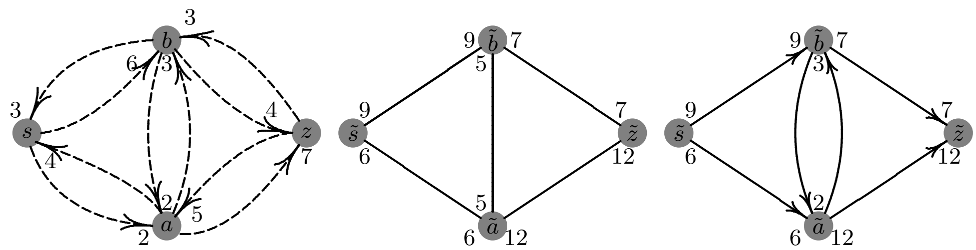


FIGURE 1. Abstract evacuation, auxiliary and reconfigured networks, respectively.

The maximum abstract flow problem is introduced in [7] and an efficient algorithm for the problem has been presented in [13]. Based on the same algorithm, the maximum abstract contraflow algorithm has been developed in [19] for the path reversal maximum abstract contraflow. The maximum abstract contraflow doubles the flow value after contraflow reconfiguration if every element in a minimum abstract cut has symmetric capacity, [19]. Kappmeier *et al.* [10] extended the abstract flow problem into maximum dynamic abstract flow problem and provided efficient solution procedure to solve the problem. To provide the solution procedure for the problem they have transformed the dynamic network into static by constructing time expanded network. For each interval, a copy of the element set $\tilde{E}$, the element set $\tilde{E}_T := \tilde{E} \times \mathbf{T}$ will be constructed. The time expanded network is constructed by $\tilde{E}_T = \{(\tilde{e}, \theta) | \tilde{e} \in \tilde{E}, \theta \in \{1, 2, \cdots, T\}\}$. The time expanded network of the dynamic abstract network is shown in Figure 2 where the numbers on the network represent transit time of the elements.

The abstract path system is allowed to be asymmetric with respect to the path capacities but the transit times are symmetric. In auxiliary network only capacities of the paths change but the transit times remain the same. As the abstract cut C_{dyn} contains an element of every temporally repeated paths, the capacity constraints are satisfied at each point of time. Thus for a subset $C_{dyn} \subseteq E_T$, the set $\gamma_t \cap C_{dyn}$ is nonempty to each $\gamma_t \in \Gamma_T$, [10]. This implies that $\sum_{\gamma_t \in \Gamma_T} f_{dyn}(\gamma_t) \leq \sum_{(e,\,\theta) \in C_{dyn}} b_e$. The number of paths created by applying the time expansion is linear in T and thus exponential in the size of input.

Example 3.2. The time expanded network of an auxiliary network can destroy the switching property. Let $(\tilde{E}, \tilde{\Gamma})$ be an auxiliary network with $\tilde{\Gamma} = \{\gamma^1, \gamma^2, \gamma^3, \gamma^4\}$ and $\tilde{E} = \{s, a, b, z\}$, where $\gamma^1 = (s;\ a;\ b;\ z)$, $\gamma^2 = (s;\ b;\ a;\ z)$, $\gamma^3 = (s;\ a;\ z)$, $\gamma^4 = (s;\ b;\ z)$ together with their reversals. The set $\tilde{\Gamma}$ satisfies the switching and order properties. For $T = 4$, we have $\Gamma_T = \{\gamma_0^1, \gamma_2^1, \gamma_0^2, \gamma_1^2, \gamma_0^3, \gamma_1^3, \gamma_2^3, \gamma_0^4, \gamma_1^4, \gamma_2^4\}$ with $\gamma_0^1 = \{(s, 0), (a, 1,), (b, 2), (z, 3)\}$ and $\gamma_1^2 = \{(s, 1), (b, 2), (a, 3), (z, 4)\}$. But time expansion of given network does not contain $\gamma_0^1 \times_{(b,2)} \gamma_1^2$ (cf. Figure 2).

Internal waiting should be allowed to construct abstract time expanded network. According to [10], the waiting has no influence on the optimality and the temporally repeated optimal solutions can be obtained even if waiting is allowed internally on time expanded network. Based on this idea, an efficient algorithm for the maximum dynamic abstract contraflow problem has been developed in [4]. The existence lexicographically maximum abstract flow and a polynomial time algorithm for the problem is given by Kappmeier [9]. In his model, the order of terminals has to fulfill compatible property if more than one terminal node is contained in a path. The existence of earliest arrival abstract flow described by Kappmeier [9] generalizes the earliest arrival flow and maximizes the dynamic abstract flow at each possible time point. For the lexicographically maximum abstract contraflow and earliest arrival abstract contraflow problems with their algorithms, we refer to [4].

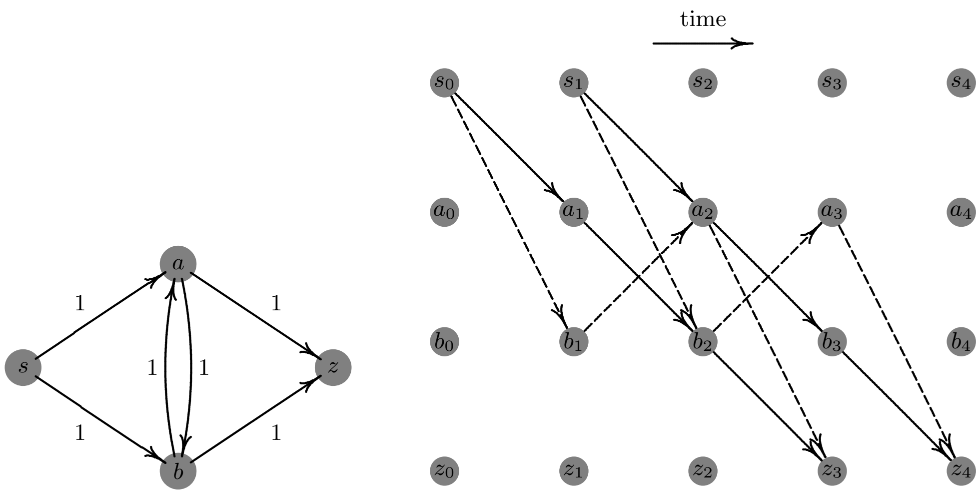


FIGURE 2. Dynamic abstract network and time expanded network, respectively.

4. Abstract Earliest Arrival Transshipment Contraflow

We study abstract earliest arrival contraflow problem on multi-terminal network. Clearly, the abstract earliest arrival contraflow cannot be obtained for multiple sinks. But the flow value can be approximated by adding supplies and demands for source and sink elements, respectively. Based on the results of [9, 17] on contraflow and abstract flow, we introduce abstract earliest arrival contraflow Problem 4.1 and propose an approximation Algorithm 4.3 to solve it. Recall that, the abstract contraflow transshipment having the earliest arrival property do not exist.

Problem 4.1. *Let $N = (G, b, \tau, S, D, T)$ be an abstract network, where $G = (E, \Gamma)$. The multi-terminal abstract earliest arrival contraflow problem is to find an abstract earliest arrival flow with path reversals capability that satisfies the supplies and demands at terminals.*

Example 4.1. Consider an abstract network with elements $E = \{s, a, b, c, z_1, z_2\}$ and paths $\Gamma = \{\overrightarrow{\gamma_1}, \overrightarrow{\gamma_2}, \overleftarrow{\gamma_1}, \overleftarrow{\gamma_2}\}$, $\overrightarrow{\gamma_1} = \{s, a, c, z_1\}$, $\overrightarrow{\gamma_2} = \{s, a, b, z_2\}$, $\overleftarrow{\gamma_1} = \{z_1, c, a, s\}$ and $\overleftarrow{\gamma_2} = \{z_2, b, a, s\}$. Suppose $d_s^+ = 12, d_{z_1}^- = 4, d_{z_2}^- = 8, d_a = 0, d_b = 0,$ and $d_c = 0$ be the balance vectors. Construct auxiliary network with $\tilde{\Gamma} = \{\tilde{\gamma_1}, \tilde{\gamma_2}\}$, $\tilde{\gamma_1} = \{\tilde{s}, \tilde{a}, \tilde{c}, \tilde{z_1}\}$ and $\tilde{\gamma_2} = \{\tilde{s}, \tilde{a}, \tilde{b}, \tilde{z_2}\}$ which is shown in Figure 3.

Besides the given feasible transshipment at time 5 there is a feasible transshipment which sends 4 units of flow on path $\tilde{\gamma}_1$ at time 4, 4 units of flow on path $\tilde{\gamma}_2$ at time 5 and another 4 units of flow on path $\tilde{\gamma}_2$ at time 6. Here, 4 units of flow arrive at time 4 because $\tilde{\gamma}_1$ has 3 transit time, another 4 units arrive at time 5 through path $\tilde{\gamma}_2$ and next 4 units arrive at time 6 through path $\tilde{\gamma}_2$. This solution sends 4 units of flows earlier, but needs more time to send the last 4 units of flows. Hence, no earliest arrival transshipment contraflow exists in the abstract network and the new solution is a 2-value-approximate flow.

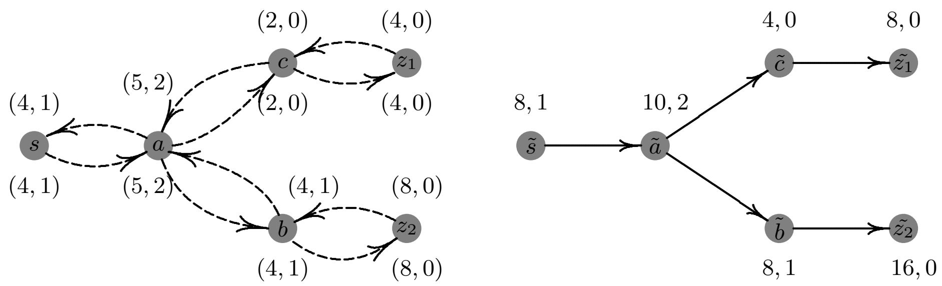


FIGURE 3. Multi-terminal abstract evacuation network and network after contraflow reconfiguration, respectively.

The proposed algorithm works on extended path system having two restrictions such that every sub-paths should preserve orders and no flow can exceed the demands and supplies, [9], as constructed below. Let (E_T, Γ_T^σ) be the abstract time expanded network for some time horizon T. We introduce additional super source elements s^*, super sink elements z^* and counting elements z_c. The time-expanded ground set is defined as: $E_T^* = E_T \cup \{s_e^* : s \in S\} \cup \{z_e^* : z \in D\} \cup \{e_c^\theta : e \in D, \theta \in \{1, 2, \cdots, T\}\}$. We extend each original temporal path $\gamma^\sigma = (e_1, e_2, \cdots, e_n) \in \Gamma_T^\sigma$ by the corresponding super terminals and a counting element. Let $s = e_1 = \text{first}(\gamma)$ and $z^\theta = e_n = \text{last}(\gamma)$ be the first and last element of γ_σ, respectively. The extended path $\gamma^{*\sigma} = (s^*, e_1, e_2, \cdots, e_n, z_c^\theta, z^*)$ contains three more elements, the super source as new first element, the super sink as last element and the counting element left of the super sink. The set of extended paths $\Gamma_T^{*\sigma}$ consists of all extended paths $\gamma^{*\sigma}$ with internal waiting at intermediate elements.

Example 4.2. Figure 4 is the extended time expanded network derived from the Figure 2.

The primal dual pair (f, y) satisfies $\sum_{e\gamma} y_e = \omega_\gamma - \lambda$ for each $\gamma \in \Gamma$ and $y_e(b_e - \sum_{e\gamma} f_e) = 0$ for each $e \in E$, where $\lambda \in Q$ is a given variable. These are the relaxed optimality conditions for weighted abstract flow and weighted abstract cut, where λ specifies deviated solutions from the optimum solution. The set of restricted elements is denoted by $R = \{e \in E : y_e > 0\} \subseteq E$. A solution of the restricted abstract maximum flow problem is a maximum flow f under the condition that the flow through restricted elements remains unchanged. The LP formulation of restricted maximum abstract flow and restricted minimum abstract cut problems presented by Kappmeier [9] are the following.

$$\max \sum_{\gamma \in \Gamma} f(\gamma) \tag{4.1}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} f(\gamma) \leq b_e, \ \forall\, e \in E \tag{4.2}$$

$$\sum_{\gamma \in \Gamma : e \in \gamma} f(\gamma) = b_e, \ \forall\, e \in R \tag{4.3}$$

$$f(\gamma) \geq 0, \ \forall\, \gamma \in \Gamma \tag{4.4}$$

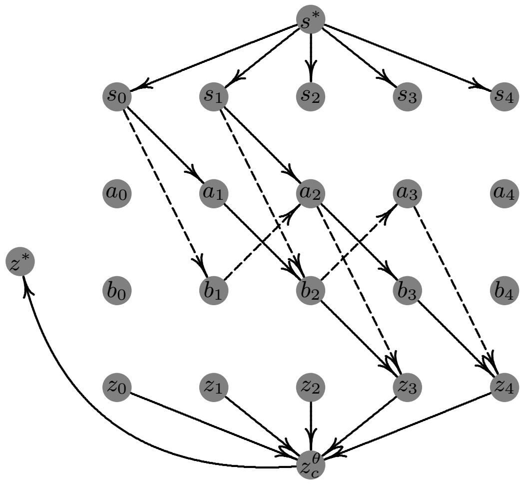


FIGURE 4. Extended time expanded network.

$$\min \sum_{e \in E} l_e b_e \tag{4.5}$$

$$\sum_{e \in \gamma} l_e \geq \omega(\gamma),\ \forall\, \gamma \in \Gamma \tag{4.6}$$

$$l_e \geq 0,\ \forall\, e \in E \backslash R, \text{ where } l_e \text{ is the length of } e \tag{4.7}$$

Let f^i be an abstract dynamic flow in the abstract time expanded network with extended paths $(A^*{}_T, \Gamma_T^{*^\sigma})$ for a time horizon i. All paths use a counting element z_c^θ for some $\theta \in \{1, 2, \cdots, i\}$. Let $v(z_c^\theta) = \sum_{\gamma : z_c^\theta \in \gamma} f_\gamma$ be the value of flow through the counting element. Define a new abstract time expanded network by specifying new capacities b'_e, where $b'_e = b_e$ for all elements e that are not counting elements. The capacity of counting elements is restricted to the flow value through them, i. e., we set $b'_{z_c^\theta} = v(z_c^\theta)$ for all counting elements z_c^θ in the time-expanded network. All paths use exactly one of the counting elements and they are all saturated. Thus, flow f^i remains feasible in the abstract time expanded network with the new capacities b'_e. Define a feasible dual solution by setting

$$y_e = \begin{cases} b'_e, & e \text{ is a counting element} \\ 0, & \text{otherwise} \end{cases} \quad \text{for all } e \in E.$$

The flow f and dual values y are feasible in the larger abstract time expanded network $(E_{i+1}^*, \Gamma_{i+1}^{*\sigma})$ for a time horizon increased by 1. The algorithm by Martens and McCormick [12] has been applied to augment flow without removing flow on the counting elements in the process of finding restricted maximum abstract flow. In the classical setting, the Triple - Optimization - Theorem, [8] states that an earliest arrival flow is equal to a minimum cost flow where the costs equal the transit times. The same idea has been applied in abstract flow model and defined rewards such that they reflect the arrival time. A path that arrives earlier has the higher reward. Paths arriving in the first time step, i. e., paths with zero travel time, have a reward of T. The reward decreases linearly with the arrival time and paths arriving at time T have a reward of 1. Let $\gamma^{*\sigma}$ be a path in the abstract time expanded network for time horizon T, θ be the arrival time of $\gamma^{*\sigma}$. Then the reward is defined as $w(\gamma^{*\sigma}) = T - \theta + 1$ which is supermodular.

Algorithm 4.3. *Approximate Abstract Earliest Arrival Contraflow Algorithm*

(1) *Given an abstract network* $N = (G, b, \tau, T, S, D)$ *with demands* d^- *and supplies* d^+.
(2) *Construct abstract auxiliary network,* $\tilde{N} = (\tilde{G}, \tilde{b}_e, T, \tilde{S}, \tilde{D})$ *with new capacity and transit time functions* $\tilde{b}(\tilde{\gamma}) = b(\overrightarrow{\gamma}) + b(\overleftarrow{\gamma})$ *and* $\tilde{\tau}(\tilde{\gamma}) = \sum_{\tilde{e} \in \tilde{\gamma}} \tilde{\tau}(\tilde{e})$.

(3) *Solve the problem in the auxiliary network using the greedy abstract value-approximate earliest arrival flow algorithm,* [9] *as follows:*

   (a) *Set* $i = 1$ *and compute an abstract flow* $\tilde{f}^1$ *in* $(\tilde{E}_1^*, \tilde{\Gamma}_1^{*\sigma})$. *Define* $\lambda = 1$.

   (b) *Let* $(\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma})$ *be the abstract time expanded network with extended paths for time horizon* $i + 1$. *Define path weights* $\omega(\tilde{\gamma}^{*\sigma}) = T - \theta + 1$. *The updated capacities and dual values are given by*

$$b'_{\tilde{e}} = \begin{cases} v(z_c^\theta), & if\ \tilde{e} = v(z_c^\theta) \text{ is a counting element} \\ b_{\tilde{e}}, & otherwise \end{cases} \quad \text{for all } \tilde{e} \in \tilde{E}_{i+1}^*,$$

$$y_{\tilde{e}} = \begin{cases} \omega(\tilde{\gamma}^{*\sigma}) - \lambda, & if\ \tilde{e} \text{ is a counting element on path } \tilde{\gamma}^{*\sigma} \in (\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma}) \\ 0, & otherwise \end{cases} \quad \text{for all } \tilde{e} \in \tilde{E}^{*i+1}.$$

   (c) *Compute an abstract flow* $\tilde{f}^{i+1}$ *in* $(\tilde{E}_{i+1}^*, \tilde{\Gamma}_{i+1}^{*\sigma})$ *applying the restricted abstract maximum flow minimum cut algorithm by Martens and McCormick* [12].

   (d) *If* $\tilde{f}^{i+1}$ *satisfies all balances, return flow value* x^{i+1}. *Else, set* $i = i + 1$ *and continue with 3b.*

(4) *A path* $\overleftarrow{\gamma} \in \Gamma$ *is reversed if and only if the flow along* $\overrightarrow{\gamma} \in \Gamma$ *is greater than* $b(\overrightarrow{\gamma})$ *or there is a non negative flow along path* $\overrightarrow{\gamma} \notin \Gamma$.

Theorem 4.4. *An approximate optimal solution for Problem 4.1 can be obtained by Algorithm 4.3.*

Proof. The Steps 2 and 3 are feasible by definition. Step 4 is well defined; i.e. not both paths $\overrightarrow{\gamma}$ and $\overleftarrow{\gamma}$ have to be switched at a time. This is ensured by the solution of the abstract flow in auxiliary network, [9]. Switching property cancels cycle flows [19], so that there is flow along $\overrightarrow{\gamma}$ or $\overleftarrow{\gamma}$ but not in both directions at the same time. Hence, the resulting flow from Step 4 is a feasible flow with path reversals in given evacuation network $N = (G, b, \tau, T, S, D)$.

Let $\tilde{N}_T^* = (\tilde{E}_T^*, \tilde{\Gamma}_T^{*\sigma})$ be the corresponding extended abstract time expanded network of $\tilde{N} = (\tilde{G}, \tilde{b_e}, T, \tilde{S}, \tilde{D})$ with intermediate waiting, [10], where T be a significantly large time horizon. Step 3 provides the 2-value-approximate abstract earliest arrival flow in the extended abstract time expanded network of auxiliary abstract network which is an optimal flow for auxiliary network, [9]. The path capacity is increased by adding the capacity of both directions between the terminals and either direction of path is allowed with modified network. A 2-value approximate earliest arrival flow with path reversal in $N = (G, b, \tau, T, S, D)$ is also a feasible solution to the earliest arrival flow problem in the auxiliary network $\tilde{N} = (\tilde{G}, \tilde{b_e}, T, \tilde{S}, \tilde{D})$. As the amount of flow sent from S to D in Step 3 is not changed in Step 4, the resulting flow is the 2-value approximate solution for Problem 4.1. □

5. Conclusion

In this paper, we discuss abstract flow, contraflow and abstract contraflow models from various papers. Integrating abstract earliest arrival flow and contraflow, we introduced abstract earliest arrival contraflow approach with discrete time settings on multi-terminal abstract network for the first time. A 2-value approximate algorithm has been proposed for the problem assuming fixed demands and supplies on sources and sinks, respectively. Our results increase the flow values at every possible time by reducing crossing conflicts with arc reversals toward the safe destinations in evacuation planning.

Acknowledgement. The authors thank to the AvH for the Research Group Linkage Program and DAAD for partnership program. The first author would like to acknowledgement UGC, Nepal for the partial support.

References

[1] N. Altay, W. G. Green, OR/MS Research in Disaster Operations Management, *European Journal of Operational Research*, 17, 475-493, 2006

[2] T. N. Dhamala, U. Pyakurel, S. Dempe, A Critical Survey on the Network Optimization Algorithms for Evacuation Planning Problems, *International Journal of Operations Research*, 15(3), 101-133, 2018

[3] Dhamala T N, Pyakurel U. Earliest Arrival Contraflow Problem on Series-parallel Graphs, *International Journal of Operations Research*, 2013, 10(1), 1-13

[4] R. C. Dhungana, U. Pyakurel, T. N. Dhamala, Abstract Contraflow Models and Solution Procedures for Evacuation Planning, *Canadian Center of Science and Education, Journal of Mathematics Research*, 10(4), 89-100, 2018

[5] R. C. Dhungana, U. Pyakurel, S. R. Khadka, T. N. Dhamala, Universally Maximum Contraflow for Evacuation Planning, *International Journal of Operations Research, Nepal*, 4, 67-78, 2015

[6] L. R. Ford, D. R. Fulkerson, Maximal Flow Through a Network, *Canadian Mathematical Society*, 8, 399-404, 1956

[7] A. Hoffman, A Generalization of Max Flow-Min Cut, *Mathematical Programming*, 6(1), 352-359, 1974

[8] J. J. Jarvis, H. D. Ratliff, Some Equivalent Objectives for Dynamic Network Flow Problems, *Management Science*, 28(1), 106-109, 1982

[9] J-P. W. Kappmeier, *Generalizations of Flows over Time with Applications in Evacuation Optimization.* PhD Thesis, Technical University, Berlin, 2015

[10] J-P. W. Kappmeier, J. Matuschke, B. Peis, Abstract Flows over Time: A First Step Towards Solving Dynamic Packing Problems, *Theoretical Computer Science, Algorithms and Computation*, 544, 74-83, 2014

[11] S. Kim, S. Shekhar, M. Min, Contraflow Transportation Network Reconfiguration for Evacuation Route Planning, *IEEE Transactions on Knowledge and Data Engineering*, 20, 1-15, 2008

[12] M. Martens, S. T. McCormick, A Polynomial Algorithm for Weighted Abstract Flow, *Integer Programming and Combinatorial Optimization*, 97-111, 2008

[13] S. T. McCormick, A Polynomial Algorithm for Abstract Maximum Flow, *In: Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 490-497, 1996

[14] U. Pyakurel, H. N. Nath, T. N. Dhamala, Partial Contraflow with Path Reversals for Evacuation Planning, *Annals of Operation Research*, DOI: 10.1007/s10479-018-3031-8, 2018

[15] U. Pyakurel, H. N. Nath and T. N. Dhamala, Efficient Contraflow Algorithms for Quickest Evacuation Planning, *Science China Mathematics*, Doi: 10.1007/s11425-017-9264-3, 2018

[16] U. Pyakurel, T. N. Dhamala, Continuous Dynamic Contraflow Approach for Evacuation Planning, *Annals of Operations Research, Springer*, 253, 573-598, 2016

[17] U. Pyakurel, T. N. Dhamala, Continuous Time Dynamic Contraflow Models and Algorithms, *Hindawi Publishing Corporation, Advances in Operations Research, Volume 2016*, Article ID 7902460, 1-7, 2016

[18] U. Pyakurel, T. N. Dhamala, Evacuation Planning by Earliest Arrival Contraflow, *Journal of Industrial and Management Optimization*, 13, 487-501, 2017

[19] U. Pyakurel, T. N. Dhamala, S. Dempe, Efficient Continuous Contraflow Algorithms for Evacuation Planning Problems, *Annals of Operation Research*, 254, 335-364, 2017

[20] S. Rebennack, A. Arulselvan, L. Elefteriadou, P. M. Pardalos, Complexity Analysis of Maximum Flow Froblem with Arc Reversal, *Journal of Combinatorial Optimization*, 29, 200-216, 2010

[21] C. Xie, D. Y. Lin, S. T. Waller, A Dynamic Evacuation Network Optimization Problem with Lane Reversal and Crossing Elimination Strategies, *Transportation Research Part E*, 46, 295-316, 2010

[22] C. Xie, M. A. Turnquist, Lane-based Evacuation Network Optimization: An Integrated Lagrangian Relaxation and Tabu Search Approach, *Transportation Research Part C*, 19, 40-63, 2011

[23] X. Zhao, Z-Y. Feng, Y. Li, A. Bernard, Evacuation Network Optimization Model with Lane-based Reversal and Routing, *Hindawi Publishing Corporation Advances in Operations Research Volume 2016*, Article ID 1273508, 2016

Dynamic flow models and algorithms for evacuation planning

TN Dhamala[1*], RC Dhungana[1] and U Pyakurel[1]

[1]*Central Department of Mathematics, Tribhuvan University, P.O.Box 13143, Kathmandu, Nepal*

*amb.dhamala@daadindia.org

Abstract

Increasing disasters worldwide have drawn the attention of researchers to the emerging field of disaster management. We consider the key models on transportation and planning networks for maximizing the flow and minimizing the cost within the limited time horizon in emergencies. Although there are varieties of models and solution techniques in evacuation planning, either analytical or simulation, it is quite challenging to develop a universally accepted solution strategy. The most acceptable optimization, simulation and heuristics on dynamic flows, abstract flows, lossy networks and facility station will be discussed in this paper. A number of earlier models and solution approaches have been generalized. This work focuses mainly on the contraflow reconfiguration strategy which plays a vital role in increasing the flow value significantly in drastically reduced time. The in depth analysis and implementation highlights will prove the efficiency of the techniques we deal with.

Keywords: Evacuation planning, Optimization, Simulation, Dynamic flow, Contraflow.

I. INTRODUCTION

Frequent natural and man-made disasters have been increasing the essence of evacuation planning. In evacuation planning the transportation network has been taken as one of the crucial issues. Management of transportation routes is one of the major problems of evacuation planning after any kind of disaster such as a hurricane, earthquake, tsunami, flooding, industrial and nuclear accident, fire, or terrorist attack. It is important, in order to safely transport as many evacuees as possible within the limited time and at low costs to safer places.

Various mathematical models proposed in a wide spectrum of fields such as engineering, mathematics, management and social sciences are in existence covering the simulation, traffic assignment, cell-transmission and optimization approaches. The typical factors usually taken into consideration by these models are, origin-destination assignment, evacuation time, modes of transportation, transportation cost, network capacity and contraflow, for instance. The evacuation parameters could be set as static, dynamic, density dependent, discrete or continuous and these decisions heavily influence the evacuation efficiency. One may represent the problem over an evacuation network where arcs or edges are the roads linking two intersecting places (nodes) of the network. The dangerous places and safe destinations are the sources and sinks, respectively.

Most of the considered objectives of the evacuation planner are, maximization of the flow, minimization of the cost and loss, minimization of the time, or a combination of these. [5] investigates the solution procedures of the dynamic flow problems with abstract flow approach. The abstract flow model introduced by Hoffman generalizes the concepts of nodes-arcs to elements-paths in the underlying network configuration.

The utmost important property, the contraflow reconfiguration in the evacuation network has been adopted in many applications and analytical solutions. We refer to [6] and the references therein for details. The advanced models, known as the lane-based routing strategy, reduces the delays by restricting turning selection options at intersections to improve the traffic flow. These routings can also be used to reduce the intersection crossing and merging conflicts.

In Section 2, we summarize basic models and analytical results of network flows in discrete and continuous time settings. Recent results on contraflow and abstract contraflow models are given in Section 3. Other solution approaches, like simulation and heuristics, and facility location-allocation are discussed in Sections 4 and 5, respectively.

II. BASIC DENOTATIONS

Let $N = (G; c_e; \tau_e; \lambda_e; S^+; S^-; T)$ be an evacuation network with a digraph $G = (V; A; \Gamma)$ having a set of vertices V, set of paths Γ and set of arcs A. The arc capacities and transit times are c_e and τ_e respectively and λ_e represents loss factor on $e \in A$. The evacuation time is represented by $\mathcal{T} = \{0, 1, \cdots, T\}$ in a discrete model, whereas it is $\mathcal{T}_c = \{[0,1), \cdots, [T, T+1)\}$ in a continuous model. A network is called abstract, if for every $\gamma \in \Gamma$ there is an order $<_\gamma$ of the elements in γ and the switching property holds in Γ. The property holds if for every $\gamma_1, \gamma_2 \in \Gamma$ and $e \in \gamma_1 \cap \gamma_2$, there exist paths $\gamma_1 \times_e \gamma_2 \subseteq \{a \in \gamma_1 : a <_{\gamma_1} e\} \cup \{a \in \gamma_2 : a >_{\gamma_2} e\}$ and

$\gamma_2 \times_e \gamma_1 \subseteq \{a \in \gamma_2 : a <_{\gamma_2} e\} \cup \{a \in \gamma_1 : a >_{\gamma_1} e\}$.

Let $f: \Gamma \to R^+$ be the static flow on an abstract network, weight $\omega(\gamma) = 1$, $\gamma \in \Gamma$ and the LP formulation:

$$\text{Maximize} \sum_{\gamma \in \Gamma} f(\gamma) \quad (1)$$

$$\text{such that} \sum_{\gamma \in \Gamma : e \in \gamma} f(\gamma) \le c_e, \text{ for all } e \in A \quad (2)$$

$$f(\gamma) \ge 0, \text{for all } \gamma \in \Gamma \quad (3)$$

The dual of the above LP can be written as:

$$\text{Minimize} \sum_{e \in A} c_e \pi_e \quad (4)$$

$$\text{such that} \sum_{e \in \gamma} \pi_e \ge 1, \quad \text{for all } \gamma \in \Gamma \quad (5)$$

$$\pi_e \ge 0, \text{for all } e \in A \quad (6)$$

The abstract flow model is the generalization of the classical network flow model of Ford and Fulkerson.

Theorem 1 For integer capacities c_e, the optimum abstract flow and abstract cut values are equal.

[5] introduced a dynamic abstract flow model by considering travel time $\tau_e \in Z^+$ for each $e \in A$ and solved it by using temporal paths in an abstract time expanded network.

Let $f_{adyn}: \Gamma_{[T,\ T+1)} \to R^+$ be the dynamic abstract flow for all $\gamma_{[t,\ t+1)} \in \Gamma_{[T,\ T+1)}$, with $\gamma_{[t,\ t+1)} =$

$$\left\{ (e, \theta) \in A_{[t,\ t+1)} \middle| e \in \gamma, \theta = [t,\ t+1) + \sum_{\gamma \in (\Gamma,\ e)} \tau(\gamma) \right\}$$

and the set of temporal paths

$\Gamma_{[T,\ T+1)} =$

$$\left\{ \gamma_{[t,\ t+1)} \middle| \begin{array}{l} \gamma \in \Gamma,\ [t,\ t+1) \in \mathcal{T}_c, \\ [t,\ t+1) + \sum_{\gamma \in (\Gamma,\ e)} \tau(\gamma) < [T,\ T+1) \end{array} \right\}.$$

The maximum dynamic abstract flow maximizes Objective (7) satisfying Constraints (8) and (9).

$$\text{Maximize} \sum_{\gamma_{[t,\ t+1)} \in \Gamma_{[T,\ T+1)}} f_{adyn}(\gamma_{[t,\ t+1)}) \quad (7)$$

$$\text{such that} \sum_{\substack{\gamma_{[t,\ t+1)} \in \Gamma_{[T,\ T+1)}: \\ (e,\theta) \in \gamma_{[t,\ t+1)}}} f_{adyn}(\gamma_{[t,\ t+1)}) \le c_e, \forall e \in A, \theta \in \mathcal{T} \quad (8)$$

$$f_{adyn}(\gamma_{[t,\ t+1)}) \ge 0, \quad \forall \gamma_{[t,\ t+1)} \in \Gamma_{[T,\ T+1)} \quad (9)$$

They prove that the maximum dynamic abstract flow equals the capacity of minimum dynamic abstract cut.

The following node-arc model introduced by [3] has been extensively researched.

$$\sum_{\sigma=\tau_e}^{T} \sum_{e \in A_v^-} f_{dyn}(e, \sigma - \tau_e) - \sum_{\sigma=0}^{T} \sum_{e \in A_v^+} f_{dyn}(e, \sigma) = 0, \quad \forall v \notin \{s, d\} \quad (10)$$

$$\sum_{\sigma=\tau_e}^{\theta} \sum_{e \in A_v^-} f_{dyn}(e, \sigma - \tau_e) - \sum_{\sigma=0}^{\theta} \sum_{e \in A_v^+} f_{dyn}(e, \sigma) \ge 0, \quad \forall v \notin \{s, d\}, \theta \in \mathcal{T} \quad (11)$$

$$\sum_{\sigma=\tau_e}^{T} \sum_{e \in A_s^-} f_{dyn}(e, \sigma - \tau_e) - \sum_{\sigma=0}^{T} \sum_{e \in A_d^+} f_{dyn}(e, \sigma) = 0 \quad (12)$$

$$0 \le f_{dyn}(e, \theta) \le c(e, \theta), \quad \forall e \in A, \theta \in \mathcal{T} \quad (13)$$

where, $A_v^+ = \{(v, w): (v, w) \in A\}$ and $A_v^- = \{(w,\ v): (w,\ v) \in A\}$ are the set of arcs leaving and entering node v, respectively. We assume that $A_d^+ = A_s^- = \Phi$. The net flow at time T is given by

$$val(f_{dyn}) = \sum_{\sigma=\tau_e}^{T} \sum_{e \in A_s^-} f_{dyn}(e, \sigma - \tau_e) - \sum_{\sigma=0}^{T} \sum_{e \in A_d^+} f_{dyn}(e, \sigma)$$

The maximum dynamic flow (MDF) maximizes the net flow at time horizon T, whereas the earliest arrival flow maximizes the flow simultaneously at every possible time point $\theta \in \mathcal{T}$ from the beginning, by maintaining the optimal solutions in earlier steps. The quickest flow problem minimizes the time horizon T to send specified flow. The transhipment problem has been developed on the basis of flows in negative time to realize former decisions. The generalized maximum dynamic problem on lossy network maximizes the amount of evacuees from the dangerous place to the safe destination within a given time horizon with minimum loss by satisfying the constraints of the generalized model, [2].

Let the function $f_{dyn}^r: A \times \mathcal{T}_c \to R^+$ be the rate of flow on an arc e, where $f_{dyn}^r(e,\ \sigma)$ denotes the rate of flow that enters arc e at continuous time t. The continuous flow models can be formulated as in the discrete flow models where the sum over time is replaced with the integrals.

$$\int_0^T \sum_{e \in A_v^-} f_{dyn}^r(e, \sigma - \tau_e) d\sigma - \int_0^T \sum_{e \in A_v^+} f_{dyn}^r(e, \sigma) d\sigma = 0 \quad \forall v \notin \{s, d\}, \sigma \in \mathcal{T}_c \quad (14)$$

$$\int_0^\sigma \sum_{e \in A_v^-} f_{dyn}^r(e, \sigma - \tau_e) d\sigma - \int_0^\sigma \sum_{e \in A_v^+} f_{dyn}^r(e, \sigma) d\sigma \ge 0 \quad v \notin \{s, d\}, \sigma \in \mathcal{T}_c \quad (15)$$

$$\int_0^T \sum_{e \in A_s^-} f_{dyn}^r(e, \sigma - \tau_e) d\sigma - \int_0^T \sum_{e \in A_d^+} f_{dyn}^r(e, \sigma) d\sigma = 0, \quad \sigma \in \mathcal{T}_c \quad (16)$$

$$f_{dyn}^r(e, \sigma) \le c(e, \sigma), \quad \forall e \in A, \sigma \in \mathcal{T}_c \quad (17)$$

The maximum dynamic flow problem in continuous time maximizes the Objective (18) satisfying the constraints (14) - (17);

$$\mathrm{val}(f^r_{dyn}, T) = \mathrm{val}(f^r_{dyn}) = \int_0^T \sum_{e \in A_s^-} f^r_{dyn}(e, \sigma - \tau_e) d\sigma = \int_0^T \sum_{e \in A_d^+} f^r_{dyn}(e, \sigma) d\sigma \quad (18)$$

The continuous time models would give more accurate results with higher computational complexity. But the discrete models can be better options for good approximations to real-life solutions.

There is a tool called natural transformation that deals with both discrete and continuous models together with the same computational complexity, in which the continuous flow is defined as follows:
$f^r_{dyn}(e, \theta) = f_{dyn}(e, \sigma)$ for time σ with $\sigma \leq \theta < \sigma + 1$.
The difference between the models depends upon whether the flow entering an arc e at time $\sigma - \tau_e$ has already arrived at the head node by σ or is still on the arc at that moment. In discrete models we assume that such a flow is already at the head node at time σ. However, in continuous time models it will reach the head node at time $[\sigma + 1)$. Thus, continuous flow is feasible and the amount of flow that can be sent from source to sink at any integer time interval $[\sigma, \sigma + 1)$, for $\sigma = 0, 1, \ldots, T - 1$, will be the same for both flows. For detail we refer to [6].

III. CONTRAFLOW RECONFIGURATION

We define a general contraflow problem and give a summary of results for both versions of the contraflow, with the reversals of arcs and paths. The contraflow configuration is established with capability of arc reversal and path reversal, separately, to increase the capacities of arcs and paths.

Problem 1 (GCFP). Given an evacuation network $N = (V; A; \Gamma; c_e; \tau_e; \lambda_e; S^+; S^-; T)$, the general contraflow problem is to find an optimal solution for a given objective with reversals of the directions from the sources to the sinks at time zero.

Contraflow with arc Reversals

To deal with contraflow with arc reversals, we define an auxiliary network $\bar{N} = (\bar{V}; \bar{A}; \bar{c}_{\bar{e}}; \bar{\tau}_{\bar{e}}; \lambda_e; S^+; S^-; T)$ with modified capacity, transit time and arcs as follows:

$$\bar{c}_{\bar{e}} = c_e + c_{e^{-1}}, \quad \bar{\tau}_{\bar{e}} = \begin{cases} \tau_e, & \text{if } e \in A \\ \tau_{e^{-1}}, & \text{otherwise} \end{cases}$$

and $\bar{e} \in \bar{A}$ if $e \in A$ or $e^{-1} \in A$, where $e^{-1} = (w, v)$ as the reversal of an arc $e = (v, w)$. The new set of paths $\bar{\Gamma}$ is defined with respect to the arc set $\bar{A}$ in $\bar{N}$. The remaining structure and data are unchanged.

Linking to the contraflow with arc reversals, the maximum dynamic, earliest arrival, lexicographically maximum and transhipment on different structures of given networks have been developed. The same idea has been extended to the generalized flow on lossy network. The generalized maximum dynamic contraflow on lossy network maximizes the gain by minimizing the possible loss between the source and the sink at any time point [2], [6].

Contraflow with path Reversals

The contraflow configuration with lane reversal deals with the path system of the network. Let $\Gamma = \{\vec{\gamma}, \overleftarrow{\gamma}\}$ be the set of paths with s − d path $\vec{\gamma}$ and d − s path $\overleftarrow{\gamma}$. The direction d − s paths have to be reversed to increase the flow value. Note that the reversal of a path is equivalent to the reversals of each element contained in the path. The reconfigured path is denoted by $\bar{\gamma}$. Let $c_{\vec{\gamma}} = \min\{c_e : e \in \vec{\gamma}\}$ and $c_{\overleftarrow{\gamma}} = \min\{c_e : e \in \overleftarrow{\gamma}\}$ be the capacities of $\vec{\gamma}$ and $\overleftarrow{\gamma}$, respectively. Then the capacity of $\bar{\gamma}$ is defined as: $c_{\bar{\gamma}} = \min\{c_{\bar{e}} : \bar{e} \in \bar{\gamma}\}$, where the capacities of paths $\vec{\gamma}$ and $\overleftarrow{\gamma}$ have been added to form a new path $\bar{\gamma}$ with new capacity $c_{\bar{\gamma}}$. But the transit time of the new path remains the same. By the construction, each path with increased capacity satisfies the switching property.

[6]extends the concept to the more general setting, where the underline contraflow network is replaced by an abstract contraflow network, where paths are reversed at time zero. The reversals are assumed to be without processing cost.

Let $N = (V; A; \Gamma; c_e; \tau_e; s; d; T)$ be a two terminal dynamic abstract network. The maximum dynamic abstract contraflow problem is to find a maximum dynamic abstract flow that can be sent from s to d in time T, if the direction of paths can be reversed at time zero. If we discard the time factor in the abstract dynamic problem, then the problem reduces to a maximum abstract flow problem. We refer to [6]for the following results.

Theorem 2 The maximum abstract contraflow problem can be solved optimally in time $O(|V| \log U)$, where U is the maximum arc capacity.

Based on natural transformation, the following result has been established.

Theorem 3 The maximum dynamic abstract contraflow in a continuous time model can be solved optimally in polynomial time complexity.

The contraflow reconfiguration of an abstract contraflow network yields the double flow value if each element in a minimum abstract cut has symmetric capacity.

We claim that the abstract contraflow on the two terminal generalized lossy network in discrete as well as continuous time settings can be solved efficiently.

IV. SIMULATION AND HEURISTICS

Accurate descriptions of traffic dynamics over evacuation networks must be recognized at the planning and operations levels. Basically, the cell transmission model has been developed in [1] to predict traffic behaviour for one link by evaluating the flow at a finite number of carefully selected intermediate points, including the entrance and exit. In this basic model, the road is divided into homogeneous cells numbered from $i = 1$ to K. All the vehicles in a cell can be assumed to advance to the next with each tick. Thus, the system's evolution obeys:

$$n_{i+1}(t+1) = n_i(t)$$

where, $n_{i+1}(t+1)$ is the number of vehicles in the cell $i+1$ at time $t+1$. It is assumed that the above recursion holds for all flows, unless traffic is slowed down by queuing from a downstream bottleneck. The "cell-transmission model," which has been based on a recursion is as follows:

$$n_{i+1}(t+1) = n_i(t) + f_i(t) - f_{i+1}(t)$$

The current conditions of flows at time t are indicated by

$$f_i(t) = \min\{n_{i-1}(t), \quad Q_i(t), \; N_i(t) - n_i(t)\}$$

where $n_{i-1}(t)$ is the number of vehicles in cell $i-1$ at time t; $Q_i(t)$ is the capacity flow into i for time interval t and $N_i(t) - n_i(t)$ is the amount of empty space in cell i at time t. The simulation would step through time, updating the cell occupancies at each tick of the clock.

Various cell transmission models have been implemented in dynamic traffic modelling techniques for an optimal no-notice mass evacuation. The prior knowledge of originating demands and zonal information has been assumed to provide the optimization formulation for a cell-based transmission for a time expanded graph.

There is a heuristic solution for multi model integrated contraflow for uncertain arrivals of evacuees in an evacuation region with low mobility population. On evacuation planning with contraflow and crossing elimination, the bi-level lane-based network optimization and simulation models have heuristic solutions.

V. FACILITY LOCATION-ALLOCATION

Location analysis is crucial in the design of networks in transportation logistics before and after any disasters. The cost for opening and operating the facilities constitutes a large share of the overall solution cost to design efficient evacuation transportation. The uncapacitated facility location is the fundamental combinatorial optimization problem addressing location decisions[4] have presented dynamic network flows and location models for evacuation planning.

Problem 2 (FlowLoc). Let $L \subseteq A$ be the feasible locations, F be all facilities, $p: F \to Z^+$ be the size of facilities and $q: L \to Z^+$ be the number of facilities that can be placed. Define $c_e^l = c_e - \max\{p(f): loc = e\}$, with $f \in F$ and $loc: F \to L$ an allocation of the facilities to the edges. The FlowLoc problem asks to determine a loc such that the $s-d$ flow in N_{loc} = (V; A; c_e^l; τ_e; s; d; T) is maximized.

We need to maintain as many $s-d$ paths as possible while placing a facility on edges for better network utilization. The single facility 1 −FlowLoc problem asks for $l = loc(f) \in L$ with size $p(f)$ such that the reduction of flow value in N is minimal. The q-FlowLoc problem asks for the locations of q facilities $f \in F$ with size $p(f)$ such that the reduction of the flow value in N is as small as possible, where at most $q(e)$ facilities can be allocated on e. The extensions of the iterative algorithm for solving 1-FlowLoc static problems to q-FlowLoc problems are not polynomial time extendable with being part of the input. An allocation of a facility of a given size reduces the capacity of the arc by the respective size.

Choosing an efficient location from the large scale evacuation of medium-sized cities is the main challenge for evacuation planning.

VI. CONCLUSIONS

Optimization and simulation methods of dynamic network flows for evacuation planning are studied. The focus is on the contraflow reconfiguration, especially, on the maximum (dynamic) abstract flows in continuous times. These results are extendable to discrete time settings as well. The widely accepted contraflow heuristics and their analytically extended studies on the so-called, lossy networks are also covered. We also discussed a combined model of facility location with logistics and dynamic network flow for evacuation planning.

ACKNOWLEDGEMENT

The authors acknowledge the DAAD partial support in this research under its partnership between TU Kaiserslautern, Tribhuvan University and Mindanao State University. The first and second authors thank University Grants Commission, Nepal for the partial support in this research.

REFERENCES

[1] Daganzo, C. F., 1994. The cell-transmission model: A simple dynamic representation of highway traffic. *Transportation Research Part B, 288(4)*, 269-281.

[2] 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.

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

[4] Hamacher, H. W., Heller, S., and Rupp, B., 2013. Flow location (FlowLoc) problems: dynamic network flows and location models for evacuation planning. *Annals of Operations Research, 207*, 161-180.

[5] Kappmeier, J. P. W., Matuschke, J. and Peis, B., 2014. Abstract flows over time: a first step towards solving dynamic packing problems. *Theoretical Computer Science, Algorithms and Computation, 544*, 74 - 83.

[6] Pyakurel, U., Dhamala, T. N. and Stephan, D., 2017. Efficient continuous contraflow algorithms for evacuation planning problems. *Annals of Operation Research*, DOI: 10.1007/s10479-017-2427-1, 1-30.

FlowLoc Problems on Evacuation Network

Ram Chandra Dhungana[1], Tanka Nath Dhamala[2]

[1,2] Central Department of Mathematics, Tribhuvan University

Correspondence to: Ram Chandra Dhungana, Email: dhungana.ramchandra@gmail.com

Abstract

The FlowLoc problem allocates the facilities in the network with minimum flow loss. Contraflow is widely accepted solution approach for evacuation planning as it maximizes the outbound capacities of roads by reversing the required road directions from risk areas to safe places. In this work, the network facility location and the contraflow approach are combined and presented some efficient algorithms to solve the integrated problem.

Keywords: Evacuation network, dynamic flow, facility location, FlowLoc problem, contraflow

1 Introduction

In recent days, people are surviving under the threat of several natural as well as man-made disasters. An essential aspect of the awareness is the choice of appropriate shelter locations to all the needy population during disasters. Nevertheless, one has to take the consideration of traffic effects and its influences on the evacuation network. Due to the importance of the best possible choice of shelter locations for minimum flow loss, their efficient planning is one of the interested areas for the recent researchers. In evacuation network, the intersections of roads are considered as nodes, road segments between nodes are considered as arcs and routes are taken as paths. The initial places where evacuees are located and start to move are considered as source nodes and the safe places where they are supposed to draw are destination (sink) nodes. Most traffic delays occur in roads due the different facilities locations around the roads. Routing people to safety with proper facility locations is an innermost challenge to manage a regional evacuation, [1]. The authors in [2] integrated firstly the network flow with location theory and open the wide window of Flowloc theory in evacuation modeling. Different contraflow models with their efficient solution algorithms have been developed after [3]. For the large scale evacuations, the evacuation time has been improved by at least 40 percent with at most 30 percent of the total arc reversals, [3].

2 Denotations Prior Works

An evacuation network $N = (A, b_e, \theta_e, S, D, T)$ is a directed graph where V, $A \subseteq V \times V$, S, D, T represent the set of vertices, arcs, sources, sinks and time horizon respectively. The capacity and travel time vectors of $e \in A$ are denoted by b_e and θ_e, respectively. The evacuation time is represented by $T_d = \{0, 1, \cdots, T\}$ in discrete model, whereas it is $T_c = \{[0, 1), \cdots, [T, T+1)\}$ in continuous model. Let $f_{dyn} : A \times T_d \longrightarrow R_{\geq 0}$ be the dynamic network flow, $A_v^+ = \{(v, w) : (v, w) \in A\}$ and $A_v^- = \{(u, v) : (u, v) \in A\}$. The node - arc formulation of dynamic flow problem with $S = \{s\}$ and $D = \{d\}$ satisfies the following constraints:

$$\sum_{\tau=\theta_e}^{T} \sum_{e \in A_v^-} f_{dyn}(e, \tau - \theta_e) - \sum_{\tau=0}^{T} \sum_{e \in A_v^+} f_{dyn}(e, \tau) = 0, \ \forall v \notin \{s, d\} \tag{1}$$

$$\sum_{\tau=\theta_e}^{\sigma} \sum_{e \in A_v^-} f_{dyn}(e, \tau - \theta_e) - \sum_{\tau=0}^{\sigma} \sum_{e \in A_v^+} f_{dyn}(e, \tau) \geq 0, \ \forall \ v \notin \{s, d\}, \sigma \in T_d \tag{2}$$

$$\sum_{\tau=\theta_e}^{T} \sum_{e \in A_s^-} f_{dyn}(e, \tau - \theta_e) - \sum_{\tau=0}^{T} \sum_{e \in A_d^+} f_{dyn}(e, \tau) = 0 \tag{3}$$

$$f_{dyn}(e, \sigma) \leq b(e, \sigma) \qquad \forall e \in A, \ \sigma \in T_d. \tag{4}$$

The maximum dynamic flow problem maximize (5) satisfying Constraints (1-4).

$$val(f_{dyn}, T) = \sum_{\tau=\theta_e}^{T} \sum_{e \in A_s^-} f_{dyn}(e, \tau - \theta_e) = \sum_{0}^{T} \sum_{e \in A_d^+} f_{dyn}(e, \tau) \tag{5}$$

The continuous flow models introduced in [4] can be formulated as in the discrete flow models where the sum over time is replaced with the integrals. Author in [5] have presented different flow problems with algorithms for evacuation planning .

Problem 1. *Suppose F be the set of all facilities, $r : F \to N$ the size of the facilities and $nol : L \to N$ the number of facilities that can be placed on the possible locations and $L \subseteq A$ the set of all feasible locations. The FlowLoc problem asks for an allocation $loc : F \to L$ of the facilities to the arcs, such that the flow value in the network $N^{loc} = (A, b'_e, \theta_e, L, r_f, S, D, T)$ is maximized, where $b'_e = b_e - \max\{r_f : loc(f) = e\}$.*

If more than one facility is placed on location l, only the size of the largest facility determines the reduction of the capacity on the edge, for detail we refer to [2].

Problem 2. *The 1-Flowloc dynamic flow problem is a 1-Flowloc problem that asks to locate the facility in possible locations in dynamic network such that flow loss is minimum in $N^{loc} = (A, b'_e, \theta_e, L, r_f, S, D, T)$.*

The location of emergency units or other supports are most affecting factors in the evacuation network. Placing any facilities on arcs can result in a smaller maximum flow value or larger flow values on the arcs. This impact has been analyzed in [2] by using the different location scenarios. If more than one facility is placed on location l only the size of the largest facility dominates the reduction of the capacity. The multi terminal q-FlowLoc problem (q-MT-FlowLoc) has been introduced and provided some heuristic solutions for the problem as the problem is NP-hard, [6].

In contraflow approach, the auxiliary network of given network will be constructed by reversing given direction of arcs without any cost whenever the flow value can be improved from source to sink. The auxiliary network $\tilde{N}^{loc} = (A, \tilde{b}_e, \theta_e = 0, L, r_f, s, d)$ is constructed from given evacuation network N as: $\overrightarrow{e} = (v, w) \in \tilde{A}$, if $\overrightarrow{e} \in A$ or $\overleftarrow{e} = (w, v) \in A$. The arc capacity function $\tilde{b}$ is given by: $\tilde{b}_e := b(\overrightarrow{e}) + b(\overleftarrow{e})$ for all arcs $\overrightarrow{e} \in \tilde{A}$ and transit time is defined as follows:

$$\tilde{\theta}(\overrightarrow{e})[= \tilde{\theta}(\overleftarrow{e})] = \begin{cases} \theta(\overrightarrow{e}) & \text{for } \overrightarrow{e} \in A, \\ \theta(\overleftarrow{e}) & \text{otherwise,} \end{cases} \quad \forall \overrightarrow{e} \in \tilde{A}.$$

The utmost important property, the contraflow reconfiguration, the analytical solutions for various contraflow problems, we refer to [7, 5, 8] and the references therein for details. The contraflow problem can be solved with the same complexity as without contraflow but flow value may be double.

3 1-FlowLoc Contraflow Problems

In this section, we propose 1-Flowloc contraflow problems on static and continuous time dynamic networks with efficient solution procedures to the problems by integrating the 1-Flowloc problems presented in [2] and contraflow problems presented in [3].

Problem 3. *The 1-FlowLoc contraflow problem in static network finds a location $l = loc(f) \in L$ for the facility f with size r_f such that the reduction of the maximum flow value is as small as possible where arc can be reversed without any cost and $\theta_e = 0 \ \forall \ e \in A$.*

Algorithm 1. *1- Flowloc with Contraflow Algorithm*

1. *Given an evacuation network $N^{loc} = (A, b_e, \theta_e, L, r_f, s, d)$ with locations L.*
2. *Construct an auxiliary network as in [3], $\tilde{N}^{loc} = (A, \tilde{b}_e, \theta_e = 0, L, r_f, s, d)$ with capacity $\tilde{b}_e = b(\overrightarrow{e}) + b(\overleftarrow{e})$.*
3. *Solve the maximum network flow problem in $\tilde{N}^{loc}$ using 1- Flowloc algorithm, [2].*
4. *A arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $b(\overrightarrow{e})$ or there is a non-negative flow along $\overrightarrow{e} \notin A$.*

Theorem 1. *Algorithm 1 solves Problem 3 optimally in strongly polynomial time.*

Problem 4. *The 1-FlowLoc contraflow problem in dynamic network finds a location $l = loc(f) \in L$ for the facility f with size r_f such that the reduction of the maximum dynamic flow value is as small as possible where arc can be reversed without any cost at time zero.*

Algorithm 2. *Dynamic Flowloc with Contraflow*

1. *Given an evacuation network $N^{loc} = (A, b_e, \theta_e, L, r_f, s, d, T)$ with locations L.*
2. *Construct an auxiliary network $\tilde{N}^{loc} = (A, \tilde{b}_e, \tilde{\theta}_e, L, r_f, s, d, T)$ as in[3].*
3. *Solve the problem in $\tilde{N}^{loc} = (A, \tilde{b}_e, \tilde{\theta}_e, L, r_f, s, d, T)$, [2].*
4. *A arc $\overleftarrow{e} \in A$ is reversed if and only if the flow along $\overrightarrow{e} \in A$ is greater than $b(\overrightarrow{e})$ or there is a non-negative flow along $\overrightarrow{e} \notin A$.*

Theorem 2. *Algorithm 2 solves Problem 4 optimally in strongly polynomial time.*

Standing on the ideas of natural transformation as in [7] and [8], generalized network flow in [5], abstract flow in [7] and 1-Flowloc model in [2], we seek to extend the 1 - Flowloc problem to the generalized lossy and abstract networks in discrete as well as continuous time settings.

4 Conclusion

In this paper, we studied both Flowloc and contraflow models from literatures. Through these models we came to know that the location for facility is the most essential force behind the efficient evacuation planning. Integrating these models, we introduced 1-Flowloc contraflow approach for the first time. We proposed efficient algorithms for 1- Flowloc static and dynamic contraflow problems in two - terminal evacuation networks.

Acknowledgment: The authors acknowledge the DAAD partial support under its partnership between TU Kaiserslautern, Tribhuvan University and Mindanao State University. The first author thank University Grants Commission, Nepal for the partial support.

References

[1] Hamacher, H.W. and Tjandra, S. A., Mathematical modeling of evacuation problems: a state of the art, in pedestrian and evacuation dynamics, (eds. M. Schreckenberger and S. D. Sharma), *Springer*, 227–266, 2001.

[2] Hamacher, H. W., Heller, S. and Rupp, B., Flow location (FlowLoc) problems: dynamic network flows and location models for evacuation planning, *Annals of Operations Research*, 207(1), 161–180, 2013.

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

[4] Philpott, A. B., Continuous-time Flows in Networks, *Mathematics of Operations Research*, 15, 640 – 661, 1990.

[5] Pyakurel, U. *Evacuation planning problem with contraflow approach*, PhD thesis, IOST, Tribhuvan University, 2016.

[6] Heller, S., and Hamacher, H. W., The multi terminal q-FlowLoc problem: *A heuristic In Lecture notes in computer science: Vol. 6701, Proceedings of the international network optimization conference (INOC 2011)*, 523—528, Berlin, Springer 2011.

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

[8] Pyakurel, U., Dhamala, T. N., Continuous time dynamic contraflow models and algorithms. *Hindawi Publishing Corporation Advances in Operations Research, Volume 2016*, Article ID 7902460, 1 – 7, 2016.

APPENDIX E

CERTIFICATES OF PARTICIPATIONS

National Conference on

Mathematics and Its Applications

(NCMA-2017)

January 11-13, 2017, Chitwan, Nepal

CERTIFICATE

This certificate is awarded to

Ram Chandra Dhungana

Tribhuvan University

for participating and presenting

a paper entitled

Flow Improvement with Fixed Switching Cost in Network Flow Models

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

TRIBHUVAN UNIVERSITY
CENTRAL DEPARTMENT OF MATHEMATICS
KIRTIPUR, KATHMANDU
NEPAL

Institute of Science and Technology

Certificate of Attendance

Ram Chandra Dhungana
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
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

Ram Chandra Dhungana

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

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

Ram Chandra Dhungana

Tribhuvan University, Kathmandu, Nepal

for participating and presenting a paper entitled *Lane Based Generalized Flows in Lossy Network for Evacuation Planning* 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

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

Ram Chandra Dhungana

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 RAM CHANDRA DHUNGANA

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

INTERNATIONAL CONFERENCE

on

Recent Advances in Informatics, Communication, Management, Health & Applied Sciences

RAICMHAS-2019

Date: February 02-04, 2019

organized by :

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, Kolkata 700 125

★ ★ ★ ★ ★

This is to certify that

RAM CHANDRA DHUNGANA

of

TRIBHUVAN UNIVERSITY

has participated/presented a paper/poster entitled

ABSTRACT FLOW LOC PROBLEMS AITH NETWORK RECONFIGURATION

in

RAICMHAS-2019

held from February 02, 2019 to February 04, 2019

S. Ghosh	Soumya Paul		K. Banerjee
Dr. Sharmistha Ghosh	Dr. Soumya Paul	Dr. Sudipta Bhattacharyya	Dr. Kaushik Banerjee
Convener	Co-Convener	Co-Convener	Co-Convener