TRAFFIC ENGINEERING WITH SEGMENT ROUTING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONIC ENGINEERING

By

Laila Tul Qadr

September 2022

TRAFFIC ENGINEERING WITH SEGMENT ROUTING
By Laila Tul Qadr
September 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Ezhan Karaşan(Advisor)

Oya Karaşan

Altan Koçyiğit

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

TRAFFIC ENGINEERING WITH SEGMENT ROUTING

Laila Tul Qadr
M.S. in Electrical and Electronic Engineering
Advisor: Ezhan Karaşan
September 2022

Segment routing in traffic engineering is a relatively new technique in the field of networking. Segment routing is a simpler form of source routing where the segments that a packet will follow are written in the header of the packet. Segments are identified using Segment IDs known as SIDs. Node SIDs and Adjacency SIDs identify different types of segments: The first one identifies the shortest-path segments and the latter identifies the non-shortest direct links between two nodes. The ingress routers direct packets towards their destinations using Equal Cost Multiple Paths (ECMPs). Recently, several solutions have been proposed for traffic engineering using segment routing. The objective in these formulations is to minimize the Maximum Link Utilization (MLU) in the network. These Mixed Integer Linear Programming (MILP) based formulations do not consider all possible paths and the Running times increase beyond a reasonable value as the number of nodes and segments increase. Considering these short-comings, we introduce new formulations and algorithms for the problem. To incorporate all segment pairs into the formulation, a path-based model K-MMILP is introduced. Moreover, a flow-based model, K-MsMILP is proposed. These formulations incorporate all Adjacency SIDs, Node SIDs, and ECMPs. Furthermore, the effect of restricting the maximum path length followed by the flow on MLU and Running time is analyzed. The proposed flow-based formulation produces optimum results for all topologies considered for each of the 20 instances using a maximum of 3 segments per end-to-end path. It also significantly reduces the Running time for all topologies. For instance, for the 16-node German Network, the Running time is reduced by a factor of 14.9 times on the average. Moreover, for the 27-node European network, the older formulation could not produce optimum results within 24 hours while 3-MsMILP produced results in 2268 seconds on average.

Keywords: Segment Routing, Node SIDs, Adjacency SIDs, Equal Cost Multiple Paths (ECMPs), Maximum Link Utilization (MLU), Mixed Integer Linear Programming (MILP), Multi Commodity Flow(MCF).

ÖZET

SEGMENT YÖNLENDIRME ILE TRAFIK MÜHENDISLIğI

Laila Tul Qadr
Elektrik ve Elektronic Mühendisliği, Yüksek Lisans
Tez Danışmanı: Ezhan Karaşan
Eylül 2022

Trafik mühendisliğinde segment yönlendirme, ağ oluşturma alanında nispeten yeni bir tekniktir. Segment yönlendirme, bir paketin izleyeceği segmentlerin paketin başlığına yazıldığı daha basit bir kaynak yönlendirme şeklidir. Segmentler, SID'ler olarak bilinen Segment Kimlikleri kullanılarak tanımlanır. Düğüm SID'leri ve Bitişiklik SID'leri, farklı segment türlerini tanımlar: Birincisi, en kısa yol segmentlerini tanımlar ve ikincisi, iki düğüm arasındaki en kısa olmayan doğrudan bağlantıyı tanımlar. Giriş yönlendiricileri, Eşit Maliyetli Çoklu Yolları (ECMP'ler) kullanarak paketleri hedeflerine yönlendirir. Son zamanlarda, segment yönlendirme kullanan trafik mühendisliği için çeşitli çözümler önerilmiştir. Bu formülasyonlardaki amaç, ağdaki Maksimum Bağlantı Kullanımını (MLU) en aza indirmektir. Bu Karışık Tamsayılı Doğrusal Programlama (MILP) tabanlı formülasyonlar tüm olası yolları dikkate almaz ve düğüm ve segment sayısı arttıkça hesaplama süreleri makul bir değerin ötesine geçer. Bu eksiklikleri göz önünde bulundurarak, problem için yeni formülasyonlar ve algoritmalar sunuyoruz. Tüm segment çiftlerini formülasyona dahil etmek için yola dayalı bir model K-MMILP tanıtıldı. Ayrıca, akış tabanlı bir model olan K-MsMILP de önerilmiştir. Bu formülasyonlar, tüm Bitişik SID'leri, Düğüm SID'lerini ve ECMP'leri içerir. Ayrıca, akışın takip ettiği maksimum yol uzunluğunu kısıtlamanın MLU ve hesaplama süresi üzerindeki etkisi analiz edilmiştir. Önerilen akış tabanlı formülasyon, uçtan uca yol başına maksimum 3 segment kullanarak 20 örneğin her biri için düşünülen tüm topolojiler için optimum sonuçlar üretir. Ayrıca tüm topolojiler için hesaplama süresini önemli ölçüde azaltır. Örneğin, 16 düğümlü Alman Ağı için, hesaplama süresi ortalama olarak $14,9$ kat oranında azaltılır. Ayrıca, 27-düğümlü Avrupa ağı için, 3-sMILP 24 saat içinde optimum sonuçlar üretemezken, 3-MsMILP ortalama 2268 saniyede sonuç üretti.

Anahtar sözcükler: Segment Yönlendirme, Düğüm SIDs, Komşuluk SID'leri, Eşit Maliyetli Çoklu Yollar (ECMPs), Maksimum Bağlantı Kullanımı (MLU), Karışık Tamsayılı Doğrusal Programlama (MILP), Çoklu Ürün Akışı (MCF).

Acknowledgement

First of all, I would like to thank Allah Almighty for always being by my side throughout this journey. This thesis would not have been possible without the constant supervision of my advisor Prof. Dr. Ezhan Karaşan. He built my technical skills, supported me in my troubles and boosted my morale to work better.

A constant support from back home of my parents, Anayat Ullah Khan Hijazi and Zari Gul, has played a vital role in the completion of my thesis. They trusted me to move to another country to continue my studies. They supported me through the thick and thin of this journey and celebrated my little achievements. Thank you for providing me with this strength.

I would also like to thank my siblings Mariam Khan, Hafiz Anwar Ullah Khan, and Hafiz Obaid Ullah Khan for being a ray of sunshine on my cloudy days, for always taking care despite us being thousand of miles apart.

Credit is also due to my friends in Pakistan, Fl Lt. Ahmed Hassaan Arshad, Capt. Furqan Hameed and Abdul Hadi Qureshi for always taking up my tantrums and supporting me throughout my time here and otherwise.

Last but not the least, my special gratitude goes out to all of my friends in Bilkent who made this place a home away from home, who made my hard days easy, easy days happy, and happy days happier. My heart will always be in Bilkent to recherish the innumerable memories I have made here.

This thesis is dedicated to my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is anticipated that the Internet traffic will increase manyfold in the upcoming years [3] in view of the Internet-of Things (IoT) [4] and real-time applications [5]. As the traffic is increasing, there has been a continuous concern for the effective management of this traffic within the constrained and limited resources of the already implemented networks. Hence, the need for effective Traffic Engineering (TE) is being continuously enhanced to dynamically utilize the network capacity [6]. In the conventional IP networks, there is a high chance of traffic congestion [7] as the traffic is being routed over the shortest-paths only. To fully utilize the network capabilities, there is a need of load balancing in the existing networks [8] which utilizes MPLS [9] and SDN [10] architectures in the backbone.

Segment routing [11] has surfaced as a new technique to route traffic in the networks to obtain optimum network performance and to achieve load balancing. Segment routing is source-routing mechanism [12] in which the path of the traffic is not chosen arbitrarily. In fact, the path of the traffic is well-defined beforehand and is added in the packet header of the IP protocol. As the traffic moves from the source node to the destination node, the segment list in packet header is used as a guiding path by the traffic. If the next node is explicitly defined in the segment list, it follows the shortest path to that particular node. In the other case, it follows a non-shortest direct link to reach the next node. When the number of segments to be used is limited, segment routing limits the paths and flows that can be used to route the traffic.

Figure 1.1 explains different SIDs for different scenarios. In all of these scenarios, Node SID always follows the shortest-path between any two nodes in the network. If an Adjacency SID is used, it follows the direct link to reach the node irrespective of the cost of that link. For the network shown in the figure, the shortest path length between source node s and destination node d is 4. In case a node-SID is used, the flow follows the shortest path $s - 0 - 3 - d$ to reach the destination. In the next example, adjacency-SID is used between $s - 1$ and node-SID is used between $1 - d$ to complete the path. When an adjacency SID is used, the flow will follow the direct link between two given nodes irrespective of the link cost. Here, $s - 1$ has link cost of 5 which the flow utilizes. Between $1 - d$, the shortest path is utilized using node-SID. For the next scenario, a combination of node-SID and adjacency-SID is used. The flow follows the shortest path $s - 0 - 2$ first. An adjacency-SID is then used for $2 - d$ to complete the path.

Another interesting phenomenon associated with Traffic Engineering is the utilization of Equal Cost Multiple Paths (ECMPs) [13]. As there may exist more than one path between any two nodes in the network with the same cost, it must be incorporated in the system that the traffic is equally split among these ECMPs. Here, it is assumed that the traffic bifurcation will not cause out-of-order packets at the destination node.

In Figure 1.2, two networks are shown where s is the source node and d is the destination node. In the first network, there exist two ECMPs from s to d: $[s, 0, 2, d]$ and $[s, 0, 3, d]$. Here, node 0 is the bifurcating node. The traffic then splits equally from the bifurcating node reaching at the destination node. For the second network in this figure, there exist 3 ECMPs from s to d: $[s, 0, 2, d]$, $[s, 0, 3, d]$ and $[s, 1, 3, d]$. In this network, there are two bifurcating nodes, s and 0. The traffic splits equally among the paths at these nodes. At node 3, the traffic from the two preceding links $\{0, 3\}$ and $\{1, 3\}$ combines to make the flow at the link $\{3, d\}$ equal 3/4.

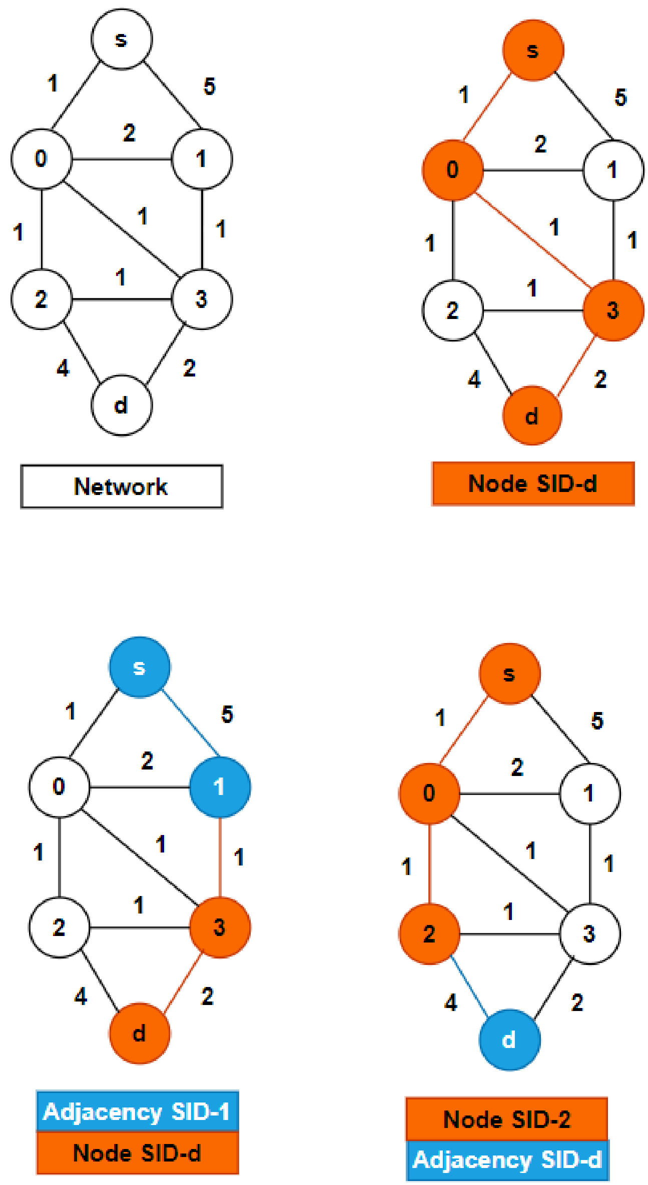

Figure 1.1: Example of Segment Identifiers (SIDs)

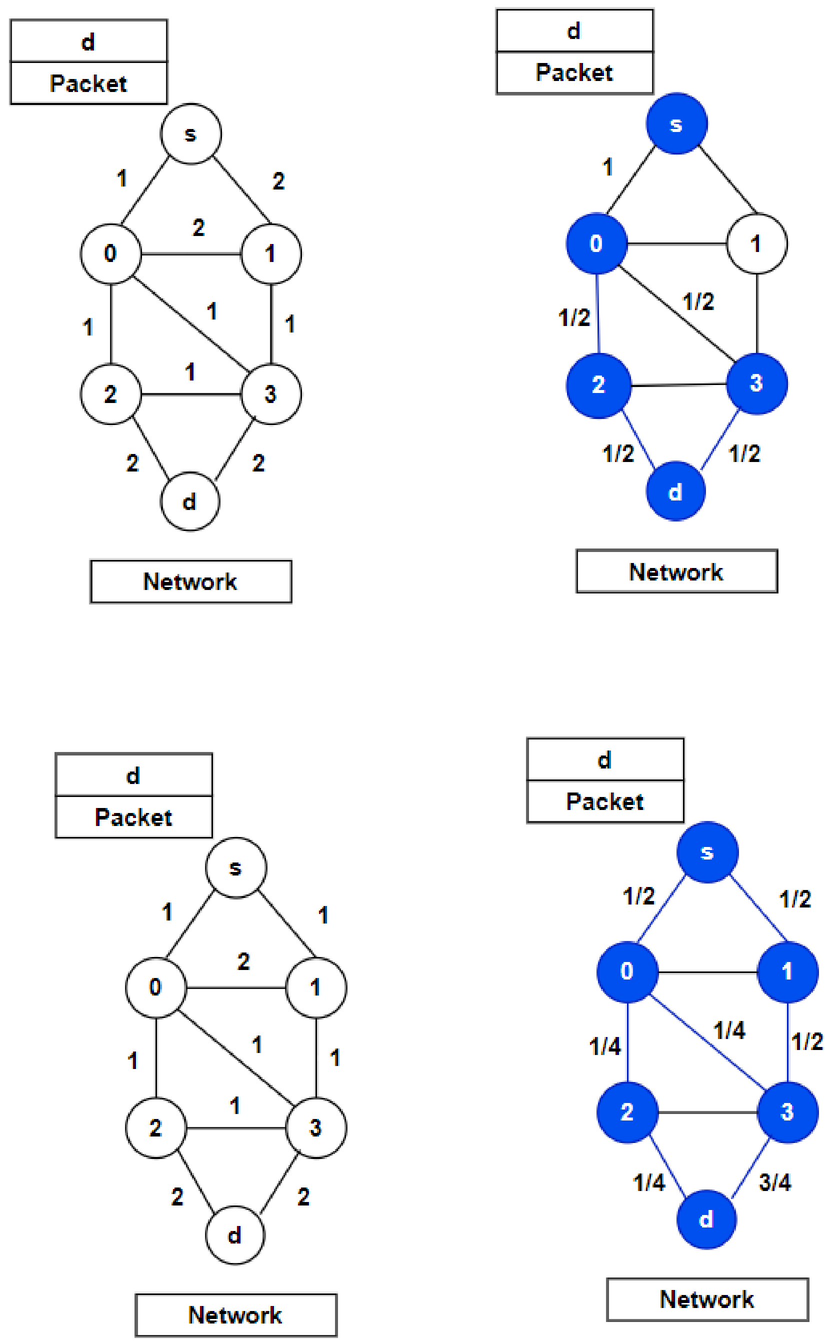

Figure 1.2: ECMP Illustration

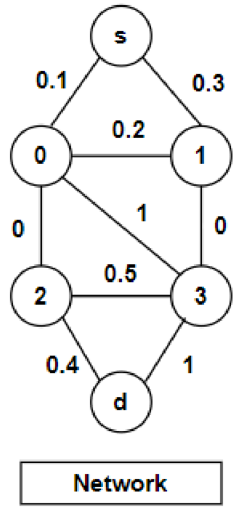

Figure 1.3: MLU Illustration

Traffic engineering in Segment Routing [14] has been explored in literature to minimize the Maximum Link Utilization (MLU) of the network. MLU is defined as the largest flow ratio in any given network for a traffic matrix. Figure 1.3 shows a network where the number written next to each link is the flow ratio. For this network, the maximum flow value is at link $2 - 3$ and the value is 0.5.

For any given traffic matrix, flows are routed in the network to find some K-segment path such that the load balancing is achieved. Different values of K can be used to reach more realistic results. However, K must not be too small to neglect the presence of valid paths. It must also not be too large to exceed the maximum limit of segment list size of the routers.

With this in view, different techniques have been used to incorporate segment routing in the networks. Initially, Integer Linear Programming (ILP) [14] has been used to minimize the Maximum Link Utilization. However, these formulations are constrained as they use Node SIDs only to form a valid path. It means that all the available segments are not fully utilized and the resulting MLU value is not optimum. Some other models have been proposed which include ECMPs in the optimization process; however, these formulations also did not use all the segments and paths available in the network yielding sub-optimum results. Moreover, different models have been formulated with varying number of segments K. As ILP-based models are dependent on the value of K, a new model for each value of K needs to be formulated. Although adjacency SIDs are incorporated into the model in [2], all direct links are not fully utilized to achieve the optimum load balancing. There has also been a drawback of longer running times prohibiting the implementation in real networks.

1.1 Motivation

Mixed Integer Linear Programming (MILP) is used in [2] to produce optimal solutions for the segment routing in traffic engineering. K-sMILP uses segment-based approach to construct valid paths for the flow to be routed. However, the time-complexity of the model increases tremendously with the increasing number of nodes, N, in the network. Also, this model does not consider all the segments and does not fully utilize the direct links in the network to route the flow. The optimal results produced by this model take very long running times. With these shortcomings, there is a need for new models which produce optimum results within a reasonable time by including all the segment-pair nodes and which also utilize direct links to route the traffic such that the Maximum Link Utilization (MLU) of the network is reduced to a minimum value.

1.2 Problem Statement

In our traffic engineering model, our aim is to minimize the Maximum Link Utilization (MLU) in the network. Another aspect of the traffic engineering problem is to be able to obtain solutions within a reasonable amount of time. As the network gets larger, the running times for the models also increase. This makes the existing models impractical for larger networks. Also, there is a large number of variables and constraints in the formulations which contribute towards the longer running times. These formulations do not fully utilize all the Equal Cost Multiple Paths (ECMPs) and segment-pairs in the network, including direct link segments, to route the traffic. This leaves a gap in the literature to explore and exploit the benefits of segment routing for traffic engineering.

1.3 Contribution

This thesis covers the segment routing in traffic engineering while focusing on minimizing the Maximum Link Utilization (MLU) of the network. While the previous works have achieved to obtain low MLU values, the running times remained enormously high specifically for larger network sizes. The previous models also did not fully utilize all the segments in the network. The new models which we have proposed not only produce better MLU results for all the networks but also substantially reduce the running times of the optimization models achieving results very close to the optimum values obtained from the Multi-Commodity Flow (MCF) model. For instance, for the 16-node German Network, the running time is reduced by a factor of 14.9 times on the average. Moreover, for the 27-node European network, the older model could not produce optimum results within 24 hours while 3-MsMILP produced results in 2268 seconds on average. The new formulations also utilize the direct links in the network to route the traffic flow in the network. The modifications introduced in our new models make them feasible for the implementation in practical networking scenarios. There has also been an additional study to analyze the effects of restricting the maximum allowed length in the network on both the MLU and the running times.

1.4 Organization

This thesis is organized as follows. Chapter 2 covers the literature review of the segment routing in traffic engineering with special emphasis on Mixed Integer Linear Programming (MILP) techniques. Chapter 3 presents two new models to optimize the Maximum Link Utilization (MLU) of the network while limiting the running time of the new models. It also includes an extension of the model by including additional constraints to study the effect of limiting the maximum path length traversed by the traffic. Chapter 4 includes all the relevant results of the new model to show its effectiveness in achieving the desired results. Chapter 5 concludes the thesis.

Chapter 2

Literature Survey

In this chapter, we will cover the basic traffic engineering and segment routing mechanics. The classic Multi-Commodity Flow (MCF) model is also discussed which will be used as a baseline reference in the thesis. There will also be a discussion about different models used in the literature to minimize the Maximum Link Utilization (MLU) for any traffic in the network. The shortcomings of the work are also discussed at the end of this chapter.

2.1 Traffic Engineering

Traffic engineering deals with the performance evaluation and optimum performance of conventional IP networks [6]. It covers both the traffic level and resource level objectives to achieve an operational network performance. It is implemented such that the capacity of network is utilized for optimum performance while ensuring that the traffic is correctly and efficiently routed across the network.

Although it remains true that traffic engineering in networks works with multiple objectives [15], it is also imperative to understand that these objectives and their corresponding standards may change with time. Therefore, traffic engineering remains a continuous process which incorporates the changing standards of

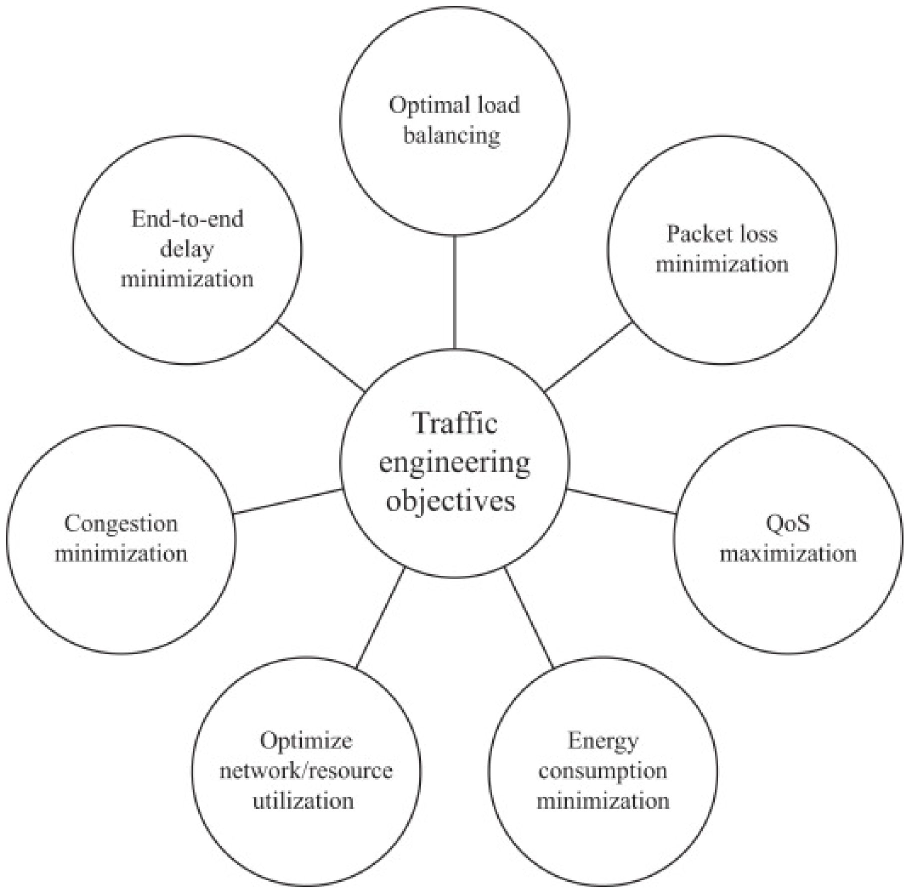

Figure 2.1: Objectives of Traffic Engineering [1]

objectives and operate accordingly.

There exist certain objectives of traffic engineering such as optimal load balancing [16]. The traffic in the network needs to be redirected from the source node to the destination node in such a way that the traffic is distributed among various paths in the network before reaching its destination. This is done to ensure that the network performs at its optimum capacity and none of the links has an overload of traffic while others are idle.

Packet loss minimization [17] is another traffic engineering objective which aims to lower down the probability of any lost data packets in the network. Traffic engineering is done in any given network to ensure that none of the data packets in the network are lost or misplaced due to congestion. In fact, it is ascertained that all the data packets reach at the destination with the minimum delays in-between.

Traffic engineering in networking also needs to maintain the maximum possible Quality-of-Service (QoS) [18]. QoS is such that as is experienced by the end user or the customer of the network. It is one of the most fundamental objectives of traffic engineering as the experience of customer is of the paramount importance. There are certain other parameters embedded in the umbrella term of QoS and includes several Key Performance Indicators (KPIs) of the network.

It is necessary in any network to ensure that the energy consumed by the network is minimized to the lowest possible value [19]. Every router in the network is constrained by its maximum energy consumption. Therefore, traffic engineering of the network must ensure that the traffic in the network is routed such that the energy is minimized to the lowest possible value.

A network must be optimized and all the resources must be utilized efficiently and it is one of the objectives of traffic engineering [20]. It means that the network must be designed and all the resources allocated to the network must be used in a way that the performance of the network is maximised and the cost is minimized.

It must also be ensured that the probability of congestion in the network is minimized [21]. As several data packets are routed in the network simultaneously, there remains a high probability that any given link in the network will experience congestion. If this happens, the data might get lost or the link breaks down. Hence, traffic engineering in the network must minimize the congestion in the network.

End-to-end delay in the network should also be minimized as an important objective of traffic engineering [22]. It means that the time taken for a packet to reach from source to destination needs to be minimized to ensure high speed and least latency in the network while maximising the user experience. This is considered as an objective of traffic engineering in networking.

2.2 Segment Routing

Segment routing [23] is a new network paradigm to optimize network performance. Its ability to be implemented on top of MPLS [24] and IPv6 [25] makes it suitable for existing networks without the need to introduce any changes in the control planes of the existing networks. In segment routing, all the nodes are identified by two types of Segment Identifiers (SIDs) [26]: Node SID and Adjacency SID.

A Node SID [27] is essentially identified by the segment with the shortest length path in the network. It is an SID that can by unambiguously used to identify the segment in the network. An Adjacency SID [28] is a direct link segment which may not be the shortest path segment. This direct link segment is also referred as non-shortest path link.

In segment routing [29], traffic can follow any arbitrary path. This path is decided by the segment list appended in the header of the packet. This list is not unique for every path in the network. There may exist multiple ways to define a path in the segment list. If only the destination node is defined in the segment list, the flow will follow the shortest path to reach the node. If the packet header contains more than one SID, the flow reaches each node using the shortest path until it reaches the final destination. In case, an adjacency SID is to be utilized, it needs to be defined in the segment list. If the packet header contains an adjacency SID, the flow follows the direct link to that node irrespective of the cost of that link. In Figure 2.2, the first diagram shows the case where $\{d\}$ is the segment list. For this, the packet follows the shortest path $s - 0 - 3 - d$ to reach destination node. In the next diagram, the segment list is defined in packet header as $\{1, d\}$. The packet first follows the shortest path to reach node 1 which is $s - 0 - 1$. Upon reaching node 1, this SID is removed from the segment list using node-popping [27]. Now the packet follows the shortest path from node 1 to node d which is $1 - 3 - d$. In the third diagram, a special case of adjacency SID in the segment list is explained. As the segment list is $\{2, < d >\}$, and $< d >$ identifies an Adjacency SID here, the packet first follows the $s - 0 - 2$ path to reach node 2. Upon reaching node 2, the packet follows the direct link $2 - d$ to reach the destination node.

2.3 Multi-Commodity Flow Formulation

Multi-commodity flow (MCF) is the basic model to route traffic between a source-destination pair in the network. In [30], a basic MCF formulation is given where arbitrary traffic splitting is possible. With this in view, it becomes impossible for any other model to perform better than MCF. Hence, this model is used as a benchmark in the traffic engineering problems in terms of minimum MLU. The aim of this formulation is to minimize the maximum link utilization (MLU) using Linear Programming (LP). The network is modeled as a weighted graph network $G(N, E)$, where N is the set of nodes and E is the set of links. The formulation is given below:

Objective

$$\min \theta \tag{2.1}$$

subject to

$$\sum_{s_e=n} f_{ij}(e) - \sum_{t_e=n} f_{ij}(e) = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} ,\forall (ij), \forall n \in N \tag{2.2}$$

$$\sum_{ij} f_{ij}(e).t_{ij} \leq \theta.c(e), \forall e \in E \tag{2.3}$$

$$0 \leq f_{ij}(e) \leq 1, \forall (ij), \forall e \in E \tag{2.4}$$

In the above formulation, [30] describes θ as the Maximum Link Utilization (MLU) and objective function (2.1) minimizes this value for the network. In constraint (2.2), $f_{ij}(e)$ is the flow variable on every link e in the network for any source-destination pair (i, j) which the traffic has to traverse. s_e is the source node whereas t_e is the terminating node of the edge e. Here, the summation

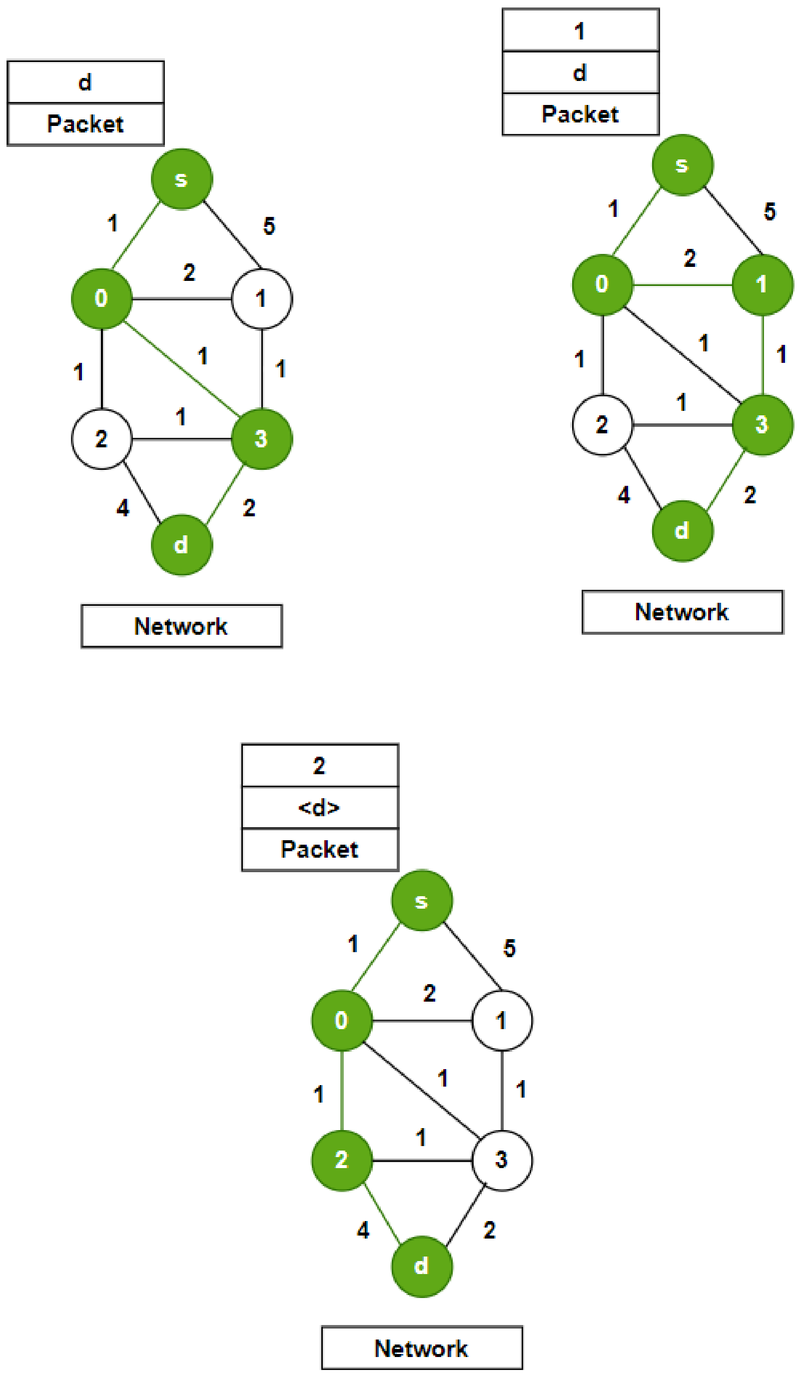

Figure 2.2: Example of Segment Lists

is done over all the source and termination node respectively. This constraint ensures that the flow is conserved for every node in the network, except for the source and destination of the flow. t_{ij} is the traffic to be routed between $i - j$ pair and $c(e)$ is the capacity of that link e. With these variables, constraint (2.3) ensures that traffic on any link remains within the capacity of that link. Constraint (2.4) ensures that the value of flow on each edge remains between 0 and 1.

The above model gives optimal solutions for MCF problem. As all the paths and flows in this model are chosen arbitrarily and there is no limit on the maximum number of segments, this model gives the ideal performance for all networks. However, as there is an inconsistent method of choosing the path of flow and the number of segments involved in this formulation, this implies that the flow problems are NP-complete [30]. For the other models in the literature and our new formulations, MCF serves as a benchmark against which the performance of all other models is gauged in terms of both MLU and running time.

2.4 K-MILP Formulation

To produce optimal solutions under segment routing, [2] proposes a Mixed Integer Linear Programming (MILP) model called K-MILP. This formulation uses K as an input parameter which is the maximum number of segments that can be used over a path. The set of segments P contains all the possible segments in the network identified by both node-SIDs and adjacency SIDs. P has two subsets P_1 and P_2. If an SID cannot unambiguously identify a segment, it falls into P_1 set. All the other segments are part of P_2. This model also includes another input variable M which is defined as the maximum total number of K-segment paths allowed. For convenience, all the parameters of this formulation are summarized in Table 2.1.

With these parameters, the formulation of K-MILP is given below:

P_1	Set of segments that cannot be unambiguously identified by single SID
P_2	Set of segments that can be unambiguously identified by single SID
P	Set of all segments in network
e_p	Binary variable, 1 if segment p contains link $e \in E$
λ	Pre-determined integer constant; $\lambda \geq 1000$
s_p	Source node of segment p
t_p	Destination node of segment p
M	Maximum number of K-segment paths
m	Path index, where $m \in \{1, 2, ..., M\}$
$x_{ij}^p\|m$	Binary variable; 1 if p is in m-th K segment path
$z_{ij}^p\|m$	Denotes fraction of flow carried on p of the m-th K segment path

Table 2.1: Input variables and common notations for K-MILP

Objective

$$\min \theta \tag{2.5}$$

subject to

$$\sum_{p \in P: s_p = n} x_{ij}^p|_m - \sum_{p \in P: t_p = n} x_{ij}^p|_m = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall m, \forall (ij), \forall n \in N \tag{2.6}$$

$$\sum_m \Big(\sum_{p \in P: s_p = n} z_{ij}^p|_m - \sum_{p \in P: t_p = n} z_{ij}^p|_m \Big) = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall m, \forall (ij), \forall n \in N \tag{2.7}$$

$$z_{ij}^{p_1}|_m - z_{ij}^{p_2}|_m \leq \lambda (2 - x_{ij}^{p_1}|_m - x_{ij}^{p_2}|_m), \forall m, \forall (ij), \forall p_1, p_2 \in P \tag{2.8}$$

$$z_{ij}^{p_1}|_m - z_{ij}^{p_2}|_m \geq \lambda (x_{ij}^{p_1}|_m + x_{ij}^{p_2}|_m - 2), \forall m, \forall (ij), \forall p_1, p_2 \in P \tag{2.9}$$

$$x_{ij}^p|_m \geq z_{ij}^p|_m, \forall (ij), \forall p \in P, \forall m \tag{2.10}$$

$$z_{ij}^p|_m > x_{ij}^p|_m - 1, \forall (ij), \forall p \in P, \forall m \tag{2.11}$$

$$\sum_m \sum_{ij} \sum_{p \in P} e_p . z_{ij}^p|_m . t_{ij} \leq \theta . c(e), \forall e \in E \tag{2.12}$$

$$\sum_{m}\sum_{p\in P|l_1\in P} z_{ij}^{p}|_m = \sum_{m}\sum_{p\in P|l_2\in P} z_{ij}^{p}|_m, \forall(ij) \tag{2.13}$$

$$\sum_{p\in P} x_{ij}^{p}|_m \leq K, \forall m, \forall(ij) \tag{2.14}$$

$$0 \leq z_{ij}^{p}|_m \leq 1, \forall(ij), \forall p, \forall m \tag{2.15}$$

$$x_{ij}^{p}|_m \in \{0, 1\}, \forall(ij), \forall p, \forall m \tag{2.16}$$

(2.5) is the objective function where θ denotes the Maximum Link Utilization (MLU) of the network. (2.6) to (2.16) are the set of constraints of the MILP problem. $x_{ij}^{p}|_m$ is the binary variable which toggles to 1 if the given segment p is used by the m-th K-segment path to route the traffic. Constraint (2.6) ensures that the flow is conserved for all nodes in all paths of the network. $z_{ij}^{p}|_m$ is the amount of flow that is carried on a segment p of the m-th K-segment path.(2.7) ensures that the given traffic is accurately split. Constraint (2.8) and (2.9) work together to ensure that for any two segments of the given K-segment path, the flow on these segments is equal. (2.10) and (2.11) assure that a segment used by the m-th K-segment path must have some flow routed on it or any segment with some flow must be used by the m-th K-segment path. e_p is a binary variable which denotes if a particular edge e in E is used by segment p. t_{ij} is the traffic to be routed between a source-destination pair ij. $c(e)$ is the capacity of each edge $e \in E$. With these variables, (2.12) ensures that the flow on any link does not exceed the capacity of that link. (2.13) is added for equal traffic splitting among ECMPs at each bifurcating node where l_1 and l_2 originate from the same node. Constraint (2.14) restricts the maximum number of segments for each path to K, taken as an input to the model.

This formulation uses a path-based approach with K number of segments per path and M number of paths for a network with N nodes. These variables make the K-MILP formulation extremely time-consuming. As shown in [2], K-MILP always yields optimum results but it takes significantly long time to produce these results.

2.5 K-sMILP Formulation

Instead of formulating a path-based solution in K-MILP, [2] proposes a simplified version of K-MILP based on segment-based model to reduce the time required for obtaining a solution. Instead of forming explicit paths in the network, K-sMILP routes traffic through a set of segments. This set of segments is then used to form a path-based solution. This reduces the time-complexity of the algorithm. The additional variables used in this model are explained in the Table 2.2.

y_{ij}^p	Denotes fraction of flow of traffic on segment p
b_{ij}^p	Binary variable; 1 if p is used by the flow
v_{ij}^p	Voltage variable for segment p traversed by traffic

Table 2.2: Additional variables for K-sMILP

K-sMILP formulation is given below:

Objective

$$\min \theta \tag{2.17}$$

subject to

$$\sum_{p \in P | s_p = n} y_{ij}^p - \sum_{p \in P | t_p = n} y_{ij}^p = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall (ij), \forall n \in N \tag{2.18}$$

$$\sum_{ij} \sum_{p \in P} e_p . y_{ij}^p . t_{ij} \leq \theta . c(e), \forall e \in E \tag{2.19}$$

$$\sum_{p \in P_1 | l_1 \in P} y_{ij}^p = \sum_{p \in P_1 | l_2 \in P} y_{ij}^p, \forall (ij) \tag{2.20}$$

$$b_{ij}^p \geq y_{ij}^p, \forall p, \forall (ij) \tag{2.21}$$

$$v_{ij}^p |_{s_p = i} = b_{ij}^p |_{s_p = i}, \forall p, \forall (ij) \tag{2.22}$$

$$v_{ij}^{p_1} |_{s_{p_1} = n} - v_{ij}^{p_2} |_{t_{p_2} = n} \geq 1 - \lambda (2 - b_{ij}^{p_1} - b_{ij}^{p_2}), \forall p_1, p_2 \in P, \forall (ij), \forall n : n \neq i, j \tag{2.23}$$

$$v_{ij}^p |_{t_p = j} \leq K + \lambda (1 - b_{ij}^p |_{t_p = j}), \forall p, \forall (ij) \tag{2.24}$$

$$0 \leq y_{ij}^p \leq 1, \forall (ij), \forall p \tag{2.25}$$

$$b_{ij}^p \in \{0,1\}, v_{ij}^p \in \mathbb{Z}^+, \forall p, \forall (ij) \tag{2.26}$$

y_{ij}^p is the fraction of flow on the particular segment $p \in P$. (2.18) ensures that the flow is conserved at every node in the network. (2.19) ensures that no link carries more flow than its capacity. Constraint (2.20) is included to make sure that the traffic is split equally among all ECMPs. b_{ij}^p is the binary variable which is toggled to 1 if the corresponding flow variable y_{ij}^p is 1 in (2.21). Constraints (2.22) to (2.24) ensure that the path of flow must not exceed K number of segments. For this reason, a new variable v_{ij}^p is included which increments its value to 1 for each segment traversed by the traffic. At the end, it is ensured that this value always remains within the upper limit of K number of segments at the destination node. Figure 2.3 explains the working of voltage variable v_{ij}^p. A constant integer λ is also introduced in this model. Its value is chosen to be $\geq$ 1000.

[2] obtains results for this model on five different topologies and observes that this model is more efficient in terms of time than K-MILP. With this model, there are some genuine paths that are discarded as the solution due to the presence of the voltage parameter, v_{ij}^p. However, it was observed that the optimal solutions were obtained for the K-sMILP model as well. In terms of time, K-sMILP was more efficient than K-MILP but the running times are still significantly longer than the running times of MCF.

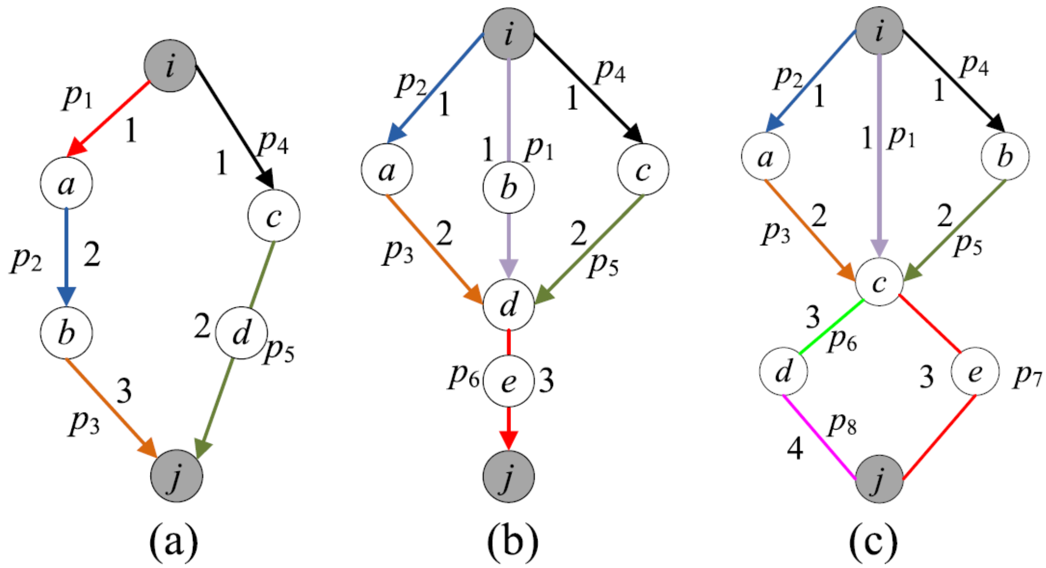

Figure 2.3: Voltage Assignment: Working of v_{ij}^{p} [2]

2.6 Shortcomings of K-MILP and K-sMILP

The literature discusses different models to route the traffic flow over the network to produce optimal solutions for Maximum Link Utilization (MLU). Multi-Commodity Flow (MCF) uses arbitrary paths to route the traffic. Moreover, it uses arbitrary number of segments making it the ideal model to route the traffic flows in the network. The routes penetrated by MCF cannot be adapted to the shortest-path routing algorithms running at the routers.

K-MILP and K-sMILP in [2] were able to produce optimal results using the Mixed Integer Linear Programming. K-MILP utilizes path-based approach to route flows in the network. However, as it is dependent on several input parameters, the time-complexity of this model increases with increasing number of nodes N in the network. K-MILP was able to produce optimal solutions but the running times for this model were extremely long.

K-sMILP is a simplified model which also utilizes Mixed Integer Linear Programming but uses segment-based approach to form a valid path for the flow.

This increases the simplicity of this model. Despite the reduction in solution space of this model to produce optimal solutions, its performance was found to be highly comparable with the performance of K-MILP. It also reduced the running time of the model to produce optimal solutions. However, the running time produced by K-sMILP were still not comparable with the running time of MCF which made it unsuitable for larger topologies in the real-time networks. Also, this model does not utilize all possible paths in the network.

With this discussion, it is evident that there is a need for a new model which utilizes all the node-pair segments and all the direct links present in the network. The new model needs to reduce the number of variables and constraints to reduce the running time so that it can be applied for reasonably sized networks.

Chapter 3

Improved Path and Flow-Based Formulation for Segment Routing

This chapter first introduces K-MMILP, the improved model of path-based formulation. An improved model of flow-based formulation K-MsMILP is also presented here. Further, the flow-based formulation is extended with the length constraint which is presented in the last part of this chapter.

3.1 K-MMILP Formulation

With a view to obtain optimal solutions for segment routing in traffic engineering, a model based on Mixed Integer Linear Programming was previously introduced. This model, called K-MILP, utilized non-shortest paths and all Equal Cost Multiple Paths (ECMPs) to obtain the results. However, the non-shortest direct links were not utilized leading to potentially sub-optimal results. Moreover, solving K-MILP was extremely time-consuming which reduced the efficiency of the model. In some cases, particularly on larger topologies, the running time was extraordinarily long [2].

In order to fully utilize the non-shortest direct links to route traffic in the network, and to reduce the running time of the model, a new formulation called Modified Mixed Integer Linear Programming, denoted by K-MMILP, is introduced. This model is formulated with a view to:

1. Utilize all non-shortest direct links in the network
2. Produce optimal results
3. Reduce the running time

K-MMILP is formulated for path-based routing. To include all the non-shortest direct paths in the network to route the available traffic, a new set P is defined. This set includes all the shortest segments including those in ECMPs. This set also has all non-shortest direct link segments. When traffic is to be routed over the network, it is routed through the shortest path available between the source-destination pair. If the ECMPs exist between the source-destination pair, the traffic flow is equally split among all the available ECMPs. For instance, if 2 ECMPs exist between any two nodes in the network, then the traffic is equally divided and routed over both paths. There also exists another option for traffic flow to follow. If the non-shortest direct link exists between two nodes, the traffic can also follow this non-shortest link to reach the destination. Moreover, the traffic can be split between the shortest and non-shortest path such that the overall link utilization is reduced to the minimum value.

To reduce the running time of the already existing model, K-MILP is modified to have fewer number of constraints to be included in K-MMILP. A simple modification to calculate the ratio of flow beforehand over the ECMPs has reduced the number of constraints. This introduces the complexity of $O(N^2.E)$ Another variation introduced is to ensure that there must be a flow on segment if it is being used. It is achieved by linking the flow variable with the binary variable. By doing so, the number of constraints have been reduced to achieve shorter running times for the model.

The model is implemented over a weighted graph denoted by $G = (N, E)$. N

is the set of nodes in the graph and E is the set of edges in the network. Every link in the weighted graph has its weight and capacity, denoted by $w(e)$ and $c(e)$ respectively. Further, a set P is defined which contains all the node-pair segments in the network. For the sake of convenience, all the sets used in the formulation are tabulated below:

N	Node set
E	Edge set
P	Node-pair segments

Table 3.1: Sets used in K-MMILP

The model uses parameters and decision variables. Parameters are given as an input to the model whereas the decision variables are to be decided after solving the formulation. All the parameters and common notations used in the model are explained in the table below:

K	Maximum number of segments
M	Maximum number of K-segment paths
m	Index of path where $m \in \{1, 2, ..., M\}$
e_p	Binary variable, 1 if direct-link segment $p \in P$ is used
g_p^e	Percentage of traffic routed over ECMP segment p on edge $e \in E$
s_p	Source node of segment $p \in P$
t_p	Destination node of segment $p \in P$
$c(e)$	Capacity of each link $e \in E$
t_{ij}	Traffic demand for a source-destination pair (i, j)

Table 3.2: Parameters and common notations for K-MMILP

The decision variables of model K-MMILP are explained in Table 3.3.

θ	Maximum Link Utilization
x_{ij}^{pm}	A binary variable, 1 if p is in m-th K-segment path for traffic t_{ij}
y_{ij}^{pm}	Ratio of flow routed over ECMP segment p in m-th K-segment path for source-destination pair (i, j)
z_{ij}^{pm}	Ratio of flow routed over non-shortest direct link segment p in m-th K-segment path for source-destination pair (i, j)

Table 3.3: Decision variables for K-MMILP

To find the percentage of flow on a link of a particular ECMP segment, i.e., g_p^e, the following method is employed:

1. Use Dijkstra algorithm [31] to find the weight of the shortest path of all segments $p \in P$
2. Find all paths for all segments $p \in P$ and store them in a list *path*[] using recursion in following manner:
    (a) Start traversing the graph from source node
    (b) Visit all the adjacent nodes of the source node and add this node to the list
    (c) Visit one node only once
    (d) Call the recursion again with this new node and continue it until the final node is reached
3. Find ECMPs using the following approach:
    (a) For all segments $p \in P$, compare the path length of each path in list *path*[] with the corresponding Weight[] list for all segments $p \in P$.
    (b) As soon as the value of path length exceeds the value of weight in Weight[], discard that path
    (c) If the path length equals the value in Weight[] list and the destination node is reached, add this path to a new list A[]

4. Sort list A[] with all ECMPs for all segments $p \in P$ in ascending order with respect to the number of nodes in ECMPs

5. Define 4 dictionaries and initiate with value of 0. These dictionaries are:

   (a) g_p^{uw} will have all the flow ratios for edge $(u, w) \in E$

   (b) d_p^{uw} will have all the edges used by the segment p for edge $(u, w) \in E$

   (c) $count_p^{ru}$ is count for the first node u of the edge for edge $(u, w) \in E$ where r is the number of ECMPs in A for segment $p \in P$

   (d) $count1_p^{rw}$ is count for the second node w of the edge for edge $(u, w) \in E$ where r is the number of ECMPs in list A[] for segment $p \in P$

6. Update g_p^{uw} to 1 for all segments p in list A[] with 1 shortest path, all the edges $(u, w) \in E$ for which d_p^{uw} is 1,

7. Traverse all paths in list A[] for all segments $p \in P$ and for all edges $(u, w) \in E$ if there exist ECMPs for a segment $p \in P$

   (a) Increment the value of $count_p^{ru}$ by the value of corresponding d_p^{uw} if u from $(u, w) \in E$ equals the node in path present in list A[]

   (b) Increment the value of $count1_p^{rw}$ by the value of corresponding d_p^{uw} if w from $(u, w) \in E$ equals the node in path present in list A[]

8. Find bifurcating nodes for the segments in the following way:

   (a) For all segments $p \in P$, all the ECMPs in list A[], number of which is denoted by r, all paths are traversed node-by-node for all edges $(u, w) \in E$

   (b) If the edge $(u, w) \in E$ exists in path in list A[] and $count1_p^{ru}$ has a value greater than 1, then the value of g_p^{uw} is changed according to the following equation:

   $$g_p^{uw} = g_p^{uw} + \frac{t}{count_p^{ru}} \tag{3.1}$$

   where t is the flow to be divided along the ECMPs

(c) If the edge $(u, w) \in E$ exists in path in list A[] but $count1_p^{ru}$ is not greater than 1, then the value of g_p^{uw} is changed according to the following equation:

$$g_p^{uw} = \frac{t}{count_p^{ru}} \tag{3.2}$$

where t is the flow to be divided along the ECMPs

(d) After every iteration of edge $(u, w) \in E$, the value of t is updated according to the following equation:

$$t = \frac{t}{count_p^{ru}} \tag{3.3}$$

With K-number of maximum segments in M paths and traffic matrix t_{ij} for all segments in set P, K-MMILP is formulated as follows:

Objective

$$\min \theta \tag{3.4}$$

subject to

$$\sum_{p \in P: s_p = n} x_{ij}^{pm} - \sum_{p \in P: t_p = n} x_{ij}^{pm} = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall m, \forall (ij), \forall n \in N \tag{3.5}$$

$$\sum_m \Big(\sum_{p \in P: s_p = n} (y_{ij}^{pm} + z_{ij}^{pm}) - \sum_{p \in P: t_p = n} (y_{ij}^{pm} + z_{ij}^{pm}) \Big) = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall m, \forall (ij), \forall n \in N \tag{3.6}$$

$$y_{ij}^{pm} + z_{ij}^{pm} \le x_{ij}^{pm}, \forall m, \forall (ij), \forall p \in P \tag{3.7}$$

$$\sum_{p \in P} x_{ij}^{pm} \le K, \forall m, \forall (ij) \tag{3.8}$$

$$\sum_m \sum_{ij} \sum_{p \in P} g_p^e y_{ij}^{pm} t_{ij} + \sum_m \sum_{ij} \sum_{p \in P} e_p z_{ij}^{pm} t_{ij} \le \theta c(e), \forall e \in E \tag{3.9}$$

$$x_{ij}^{pm} \in \{0, 1\}, \forall (ij), \forall p, \forall m \tag{3.10}$$

$$y_{ij}^{pm} \geq 0, \forall (ij), \forall p, \forall m \tag{3.11}$$

$$z_{ij}^{pm} \geq 0, \forall (ij), \forall p, \forall m \tag{3.12}$$

In the above formulation, θ denotes the Maximum Link Utilization. (3.4) gives the objective function of the formulation which is to minimize the maximum link utilization in the network. In constraint (3.5), x_{ij}^{pm} is a binary variable. It is introduced to denote whether a a particular segment p is used by path m to route the traffic t_{ij}. If the traffic is routed on a segment of a path for a given source-destination pair, the binary variable x_{ij}^{pm} becomes 1 for such case. In all other cases, it is 0. (3.5) ensures that the flow is conserved at all nodes in the network. y_{ij}^{pm} and z_{ij}^{pm} are the actual flow ratios on a given segment for a particular path and source-destination pair. y_{ij}^{pm} denotes the flow on the ECMP segment whereas z_{ij}^{pm} represents flow on direct link segment. Constraint (3.6) ensures that the cumulative values of both kinds of flow i.e. flow routed on ECMP path and flow routed on non-shortest direct link segment, is 1 for source node, -1 for destination node and 0 for all the other nodes in the network. This is to preserve the flows in the network for all source-destination pairs. Constraint (3.7) is included in the formulation to make sure that if a particular segment is used by the traffic, then this segment must be used by some path in the network. t_{ij} is the traffic demand to be routed between a particular source-destination pair. K is the maximum number of segments in the path. With constraint (3.8), it is ensured that the maximum number of segments are restricted by an upper value of K, given as an input to the model. e_p is calculated beforehand. It is 1 if the direct link e is used by the segment and is 0 otherwise. g_p^e is the percentage of flow carried on a link of a particular ECMP segment. Constraint (3.9) guarantees that the flow on any link e in the network is always less than the capacity of that link.

3.2 K-MsMILP Formulation

K-MMILP uses a path-based approach to route traffic over the network. The model utilizes M, i.e., the maximum number of K-segment paths to route traffic flow t_{ij} in the network. Here, a new model based on flow-based approach is presented. The Modified simplified Mixed Integer Linear Programming formulation is denoted by K-MsMILP. This model does not depend on M. Instead, an alternate approach based on flows is utilized here. The path is not constructed explicitly but depends on the segments which are used to carry the traffic flow t_{ij}. Moreover, given the complexity of K-MMILP, there was a need to make the formulation with potentially shorter running times. In this view, K-MsMILP is formulated which is the simplified version of K-MMILP and utilizes a flow-based approach.

Owing to the differences in K-MMILP and K-MsMILP explained previously, there are some changes introduced in this formulation. The sets used for both models are same as explained in Table 3.1. K-MsMILP uses the same parameters as explained in Table 3.2, except that it does not take maximum number of K-segment paths, M as an input. The decision variables for K-MsMILP are different and are explained in the table below:

θ	Maximum Link Utilization
y_{ij}^{pq}	Ratio of flow routed over ECMP segment pair $(p,q) \in P$ for source-destination pair (i,j)
z_{ij}^{pq}	Ratio of flow routed over ECMP segment pair $(p,q) \in E$ for source-destination pair (i,j)
x_{ij}^{pq}	A binary variable, 1 if segment $(p,q) \in P$ carries flow t_{ij}
d_{ij}^{k}	Maximum number of segments from source i, to node k for source-destination pair (i,j)

Table 3.4: Decision variables for K-MsMILP

In this formulation, a new variable d_{ij}^k is introduced. This variable is included to ensure that the paths constructed using the flow-based formulation stays within the limit of maximum number of segments, K.

K-MsMILP is implemented using the same weighted graph as K-MMILP, denoted by $G(N, E)$. For a given traffic demand t_{ij}, maximum number of segments K, set of all node-pair segments P and set of all edges E, K-MsMILP is formulated as follows:

Objective

$$\min \theta \tag{3.13}$$

subject to

$$\sum_{pq \in P:p=n} (y_{ij}^{pq} + z_{ij}^{pq}) - \sum_{pq \in P:q=n} (y_{ij}^{pq} + z_{ij}^{pq}) = \begin{cases} 1, & \text{if } i = n \\ -1, & \text{if } j = n \\ 0, & \text{otherwise} \end{cases} , \forall (ij), \forall n \in N \tag{3.14}$$

$$y_{ij}^{pq} \leq x_{ij}^{pq}, \forall pq \in P, \forall (ij), \text{if} pq \notin E \tag{3.15}$$

$$y_{ij}^{pq} + z_{ij}^{pq} \leq x_{ij}^{pq}, \forall pq \in P, \forall (ij), \text{if} pq \in E \tag{3.16}$$

$$\sum_{ij} \sum_{pq \in P} g_{pq}^{e} y_{ij}^{pq} t_{ij} + \sum_{ij} \sum_{pq \in P} z_{ij}^{pq} t_{ij} \leq \theta c(e), \forall e \in E \tag{3.17}$$

$$d_{ij}^{l} \geq d_{ij}^{k} + 1 - (K+1)(1 - x_{ij}^{kl}), \forall kl \in P, \forall (ij) \tag{3.18}$$

$$d_{ij}^{k} \leq K, \forall k \in N, \forall (ij) \tag{3.19}$$

$$x_{ij}^{pq} \in \{0, 1\}, \forall (ij), \forall pq \in P \tag{3.20}$$

$$y_{ij}^{pq} \geq 0, \forall (ij), \forall pq \in P \tag{3.21}$$

$$z_{ij}^{pq} \geq 0, \forall (ij), \forall pq \in E \tag{3.22}$$

$$d_{ij}^{k} \geq 0, \forall k \in N, \forall (ij) \tag{3.23}$$

In the above formulation, θ denotes the Maximum Link Utilization. (3.13) gives the objective function of the formulation which is to minimize the maximum link utilization in the network. y_{ij}^{pq} and z_{ij}^{pq} are the flow variables for ECMP segments $pq \in P$ and direct link segments $pq \in E$ respectively for source-destination pair (i, j). Constraint (3.14) thus ensures flow conservation in the network. x_{ij}^{pq} is a binary variable and has a value of 1 if a particular segment $pq \in P$ has a flow routed over it for (i, j). Similar to constraint (3.4), (3.15) ensures that a segment utilized must have a traffic flow on it if that particular segment $pq \notin E$ i.e. in the set which includes all the direct edges. (3.16) is a special case of the previous constraint where the segments pq included in E are catered and it is ascertained that if such segments are utilized, there must be some flow routed on them. g^e_{pq} is calculated beforehand similar to how it was calculated for K-MMILP. t_{ij} is traffic demand for a node-destination pair (i, j). Constraint (3.17) is then added to keep the actual flow on any particular edge $e \in E$ within its capacity. In other words, no link in the network should carry a flow more than its capacity. The variable d^k_{ij} is the maximum number of segments that the traffic has traversed from source node i to any node, k, in the node set, N. Constraint (3.18) calculates this value for all nodes using the binary variable x^{pq}_{ij}. (3.19) then ensures that the maximum number of segments while routing any traffic on the network does not exceed the value K.

To understand the working of d^k_{ij}, following example is included. The numbers written next to each link is the weight of that link.

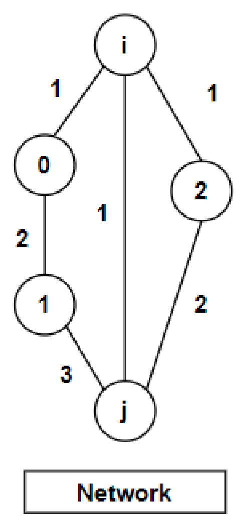

Figure 3.1: Working of d^k_{ij}: An Example

For the above graph, i is the source node and j is the destination node. All the edges are considered as individual segments. For the maximum number of segments, $K = 2$, following three cases will be considered to understand the working of d_{ij}^k:

1. Ideal case when $K \leq 2$ and all segments in graph are used, i.e., $x_{ij}^{pq} = 1$.

2. Anomalous case when $K \geq 2$ and all segments in graph are used, i.e., $x_{ij}^{pq} = 1$.

3. Case when some segments are not used, i.e., $x_{ij}^{pq} = 0$ for some segments in the graph.

Case 1: $K \leq 2$:

For this case, we will first assume that only direct link segment $i - j$ is used for flow. As only this segment is used so the value of binary variable for $i - j$ node pair, and $i-j$ segment, x_{ij}^{ij} will equal to 1. All other values of binary variable will be 0. The aim is to ensure that d_{ij}^j does not exceed $K = 2$, the maximum number of segments. Initially the value of d_{ij}^k will be 0 for all segments. For $x_{ij}^{ij} = 1$, $d_{ij}^i = 0$. d_{ij}^j will be incremented by 1 in constraint (3.18). In the following figure, the value of d_{ij}^k is introduced next to each segment depending on the segment being utilized, i.e., on the value of x_{ij}^{pq}.

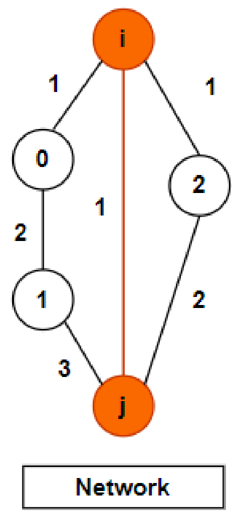

Figure 3.2: Working of d_{ij}^k: Segment $i-j$ is used only

Next, we will assume that only segment $i-2-j$ is used for flow. The value of binary variable x_{ij}^{i2} and x_{ij}^{2j} will be 1. All other values will be 0. First constraint (3.18) will increment the value of d_{ij}^2 to 1. In the next step, d_{ij}^j will be incremented to 2 which is the extreme condition.

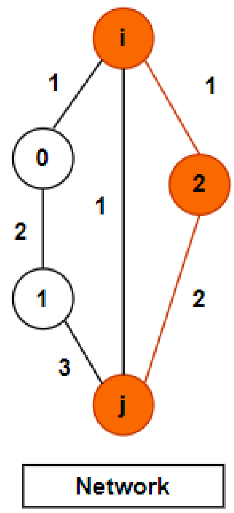

Figure 3.3: Working of d_{ij}^k: Segment $i-2-j$ is used only

Finally, we will consider that both segments, i.e., direct link segment $i-j$ and the segment $i-2-j$ are used for the flow. The corresponding binary variables for these segments will be 1 for source-destination pair $i-j$. First, for direct link segment d_{ij}^j will be incremented to 1. Next, d_{ij}^2 will be incremented to 1 for segment $i-2-j$. Finally, d_{ij}^j will be incremented to 2 using the value of d_{ij}^2. However, d_{ij}^j has a value of 1 for direct link segment $i-j$ and it has the value of 2 for segment $i-2-j$. In this scenario, the higher value of d_{ij}^j will be its final value.

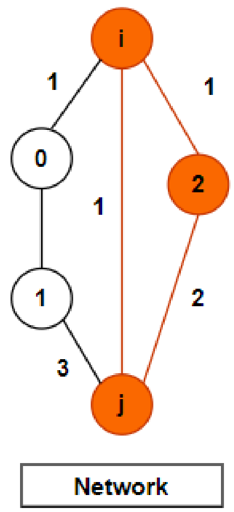

Figure 3.4: Working of d_{ij}^k: Both segments $i-j$ and $i-2-j$ are used

Case 2: $K \geq$ **2**

In this case, we will consider that the segment $i-0-1-j$ is used to route the traffic. The corresponding binary variable for segments is toggled to 1. In the first step d_{ij}^0 will be incremented to 1. In the next step, based on the value of d_{ij}^0, d_{ij}^1 will be incremented to 2. However, as the algorithm moves towards the next step, d_{ij}^j will have a value of 3 which will be prevented by constraint (3.19). In this manner, this segment will not be considered for the flow to be routed.

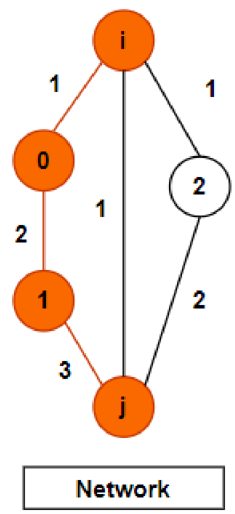

Figure 3.5: Working of d_{ij}^k: Segment $i - 0 - 1 - j$ is used

Case 3: Unused segments, $x_{ij}^{pq} = 0$
Here, we will look how the value of d_{ij}^k will be affected for unused segments in the graph. As in both of the above cases, there are some unused segments in the graph for which the binary variable x_{ij}^{pq} will assume the value 0. As the binary variable is 0 for such segments, the value of d_{ij}^k will be negative. However, as the constraint (3.23) limits the lowest value to 0, the value of d_{ij}^k will be always 0 for all the unused segments in the graph.

3.3 K-MsMILP Formulation with length Constraint

In the flow-based routing formulation, the traffic traversed through both shortest paths and non-shortest direct links. Here, we have introduced an additional set of constraints to restrict the maximum length of the path traversed by the traffic in relation to the shortest path length. The idea is to study the effect of limiting the maximum path length so that longer paths utilized by traffic engineering do not significantly increase the length. By changing the length factor, we are actually studying the effect of limiting the path length to the shortest path cost. The effect of this restriction on MLU and the running time is studied. To simplify it, we have implemented the following equation:

$$len_{max} = \alpha len_{min}$$

where,

$$len_{max} = \text{maximum path length},$$

$$\alpha = \text{length factor } \alpha \geq 1,$$

$$len_{min} = \text{shortest path length}$$

The additional variables introduced to implement the length constraint are explained in the table below:

len_{ij}^{pq}	Shortest distance if $y_{ij}^{pq} \geq 0$ and direct link length if $z_{ij}^{pq} \geq 0$
c_{pq}	Shortest distance between any two segments pq in P
w_{pq}	Weight of link pq in E
μ	A large constant integer
h_{ij}^{pq}	Indicator variable for z_{ij}^{pq}
l_{ij}^{q}	Maximum length of path traversed from source node i, to any node q in N
α	length factor
$\triangle_{ij}$	Cost of the shortest path between any two nodes i and j in N

Table 3.5: Variables for K-MsMILP with length constraint

Following constraints are added to the existing K-MsMILP formulation:

$$len_{ij}^{pq} \geq c_{pq} - \mu(1 - x_{ij}^{pq}), \forall pq \in P, \forall(ij) \tag{3.24}$$

$$len_{ij}^{pq} \geq w_{pq} - \mu(1 - h_{ij}^{pq}), \forall pq \in E, \forall(ij) \tag{3.25}$$

$$h_{ij}^{pq} \geq z_{ij}^{pq}, \forall pq \in E, \forall(ij) \tag{3.26}$$

$$h_{ij}^{pq} \leq \mu(z_{ij}^{pq}), \forall pq \in E, \forall(ij) \tag{3.27}$$

$$l_{ij}^{q} \geq l_{ij}^{p} + len_{ij}^{pq} - \mu(1 - x_{ij}^{pq}), \forall pq \in P, \forall(ij) \tag{3.28}$$

$$l_{ij}^{q} \leq \alpha \triangle_{ij}, \forall q \in N, \forall(ij) \tag{3.29}$$

$$h_{ij}^{pq} \in \{0, 1\}, \forall pq \in E, \forall(ij) \tag{3.30}$$

$$len_{ij}^{pq} \geq 0, len_{ij}^{pq} \in \mathbb{Z}^{+}, \forall pq \in P, \forall(ij) \tag{3.31}$$

$$l_{ij}^{q} \geq 0, \forall q \in N, \forall(ij) \tag{3.32}$$

Here, len_{ij}^{pq} is the shortest distance between any two segments p and q if the flow is routed using ECMPs and is the direct link length between p and q if the direct link is used to route the traffic. c_{pq} is the shortest distance between p and q if ECMPs are used to route the traffic. μ is a large constant integer value. Constraint (3.24) is used to calculate the shortest distance for the segments where traffic utilizes ECMPs. w_{pq} is the link length if a segment pq is a direct link. The variable h_{ij}^{pq} is a binary variable which is used to convert the decimal flow values in variable z_{ij}^{pq} to a higher integer value. In simple words, for any value of z_{ij}^{pq} greater than 0, the corresponding h_{ij}^{pq} is 1. If the value of z_{ij}^{pq} equals 0, then the corresponding h_{ij}^{pq} is 0 as well. In simple words, following equation relates z_{ij}^{pq} and h_{ij}^{pq}:

$$h_{ij}^{pq} = \begin{cases} 1, & \text{if} z_{ij}^{pq} > 0 \\ 0, & \text{if} z_{ij}^{pq} = 0 \end{cases}, \forall pq \in P, \forall(ij) \tag{3.33}$$

(3.26) ensures that h_{ij}^{pq} is always greater than or equal to z_{ij}^{pq}. Constraint (3.27) is included to restrict h_{ij}^{pq} to a value not greater than 1. Here, μ is a large constant integer value to round the decimal values of z_{ij}^{pq} to the value of 1. Constraint (3.25) is written for the shortest distance for the segments where traffic utilizes a direct link i.e. the segment pq is in E. l_{ij}^{q} is the maximum length of path from source node i, to any node q in the network. (3.28) calculates this maximum length of path whereas this maximum length is then restricted to the main length constraint in equation (3.29). In constraint (3.29), α is the length factor which is reduced from a maximum value to the value of 1 and $\triangle_{ij}$ is the shortest length between any two nodes i and j in N.

Chapter 4

Computational Results

This chapter covers all the results obtained for the K-MsMILP formulation and K-MsMILP formulation with the length constraint. Both formulations are studied in terms of Maximum Link Utilization (MLU) and running time. The results are obtained over 4 different topologies. Results are also compared using different topologies for both formulations.

4.1 Performance Metric

There are two performance parameters on which we have based the efficacy of our formulations. These are performance ratio and the running time. The solutions from Multi-Commodity Flow (MCF) are taken as a benchmark. MCF utilizes arbitrarily chosen paths so all the solutions obtained from MCF are the best solutions possible for that network. All the other solutions are then seen in comparison to the solutions of MCF. The comparison is drawn using the following formula:

$$\Theta = \frac{\theta_X}{\theta_{MCF}} \tag{4.1}$$

Here,

$$\Theta = \text{Performance ratio,}$$

$$\theta_X = \text{MLU of a topology obtained using the given model}$$

$$\theta_{MLU} = \text{Corresponding MLU obtained using MCF}$$

The value of Θ can never be less than 1. If it is equal to 1, the performance of the formulation is ideal.

The results are obtained using 4 topologies which are SmallNet [32], NSFNET [33], German [34] and European [35] Networks. The information about these topologies is given in the table below:

Topology	$\|N\|$	$\|E\|$
SmallNet	10	44
NSFNET	14	42
German	17	52
European	27	108

Table 4.1: Toplogies and their node-edge information

All the results are obtained for 20 instances except for the European Topology. The results for the European Topology are obtained for 10 instances due to the size of the topology. The traffic is simulated by choosing a random number in interval [1, 10]. The link costs are also chosen in a similar manner. The capacity of each link is then calculated by routing the traffic tentatively on the paths using the shortest path routing. Once the traffic is flown in this manner, the link with the highest value of traffic is identified. This highest value of traffic is then chosen as the capacity of all links in the topology. All the results are obtained on a standard laptop computer with Intel Core i7 CPU.

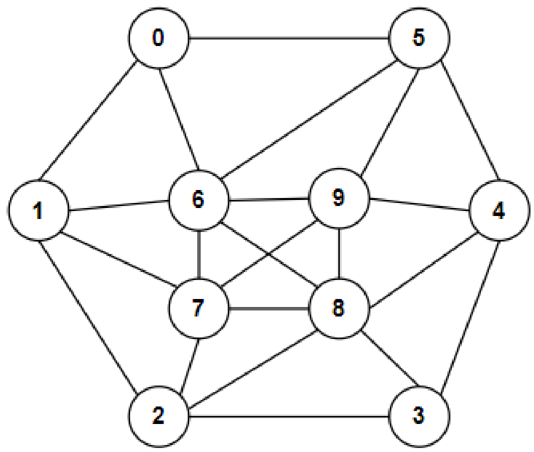

Figure 4.1: SmallNet Topology

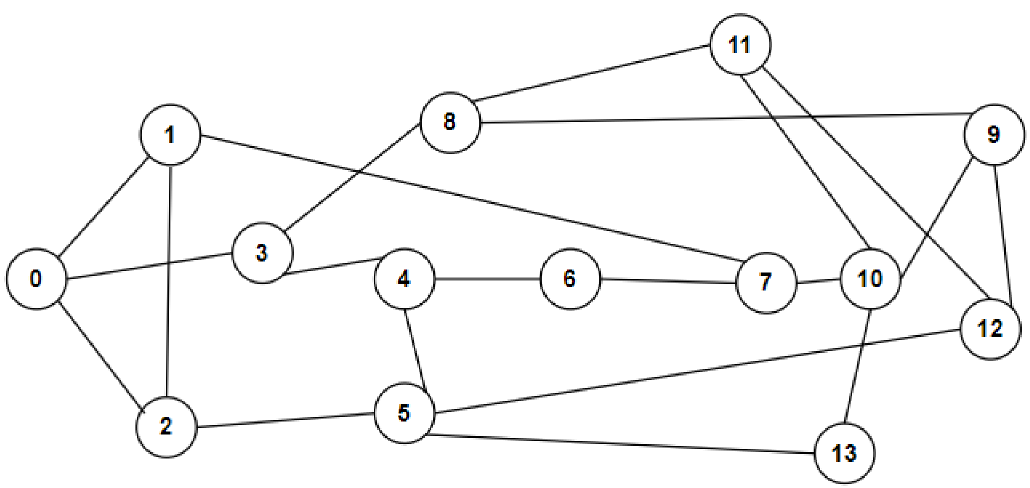

Figure 4.2: NSFNET Topology

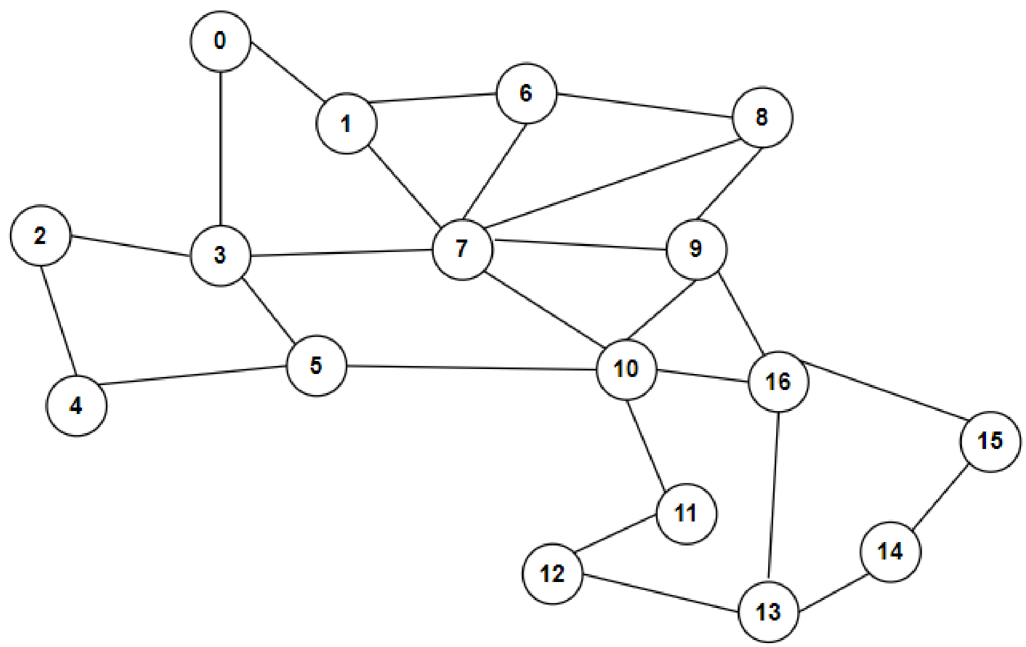

Figure 4.3: German Topology

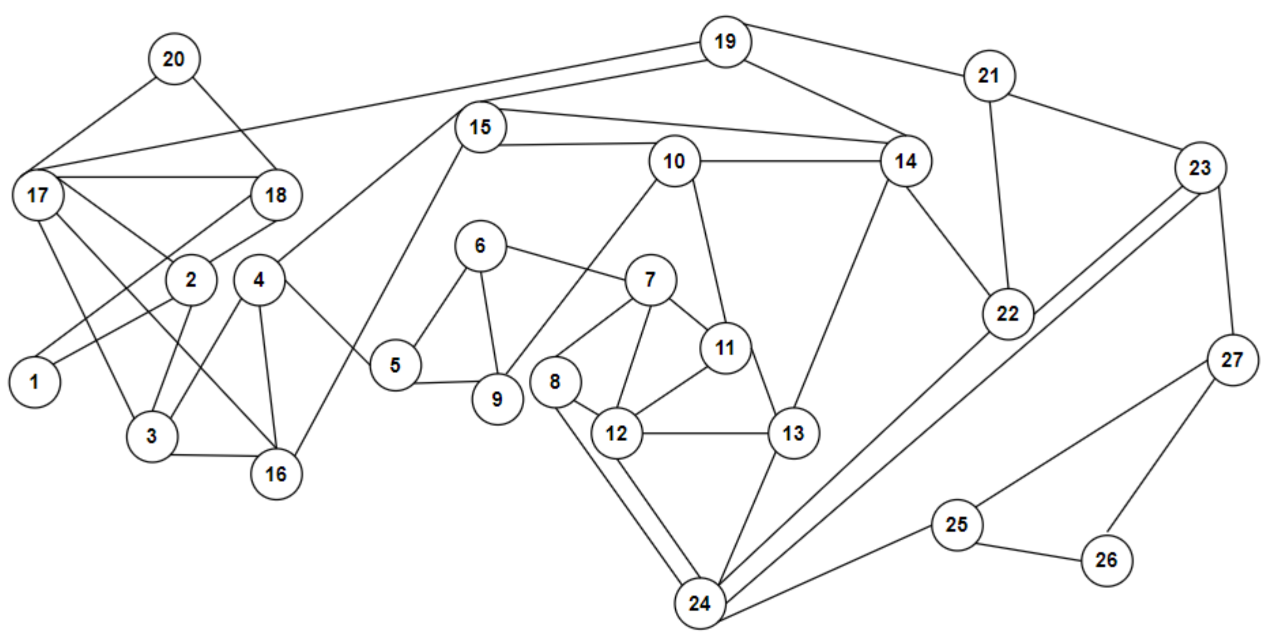

Figure 4.4: European Topology

4.2 Gurobi Optimization Solver

Gurobi [36] is a mathematical solver used to solve optimization problems. It works with some external software such as MATLAB, Python and C++. In our models, we have used Gurobi and integrated it with Python 3.8. It is easy to use as the model needs to be simply input to the solver. It works for several optimization problems such as linear constraints and bound constraints. Our usage solely relied on linear constraints.

We have used Gurobi optimization solver for our formulations to reach the optimum results for different networks. The optimum results and the time consumed by the solver are reported at the end of the simulation. Based on the constraints of the linear programming problem, Gurobi works to produce the optimum results to achieve the desired objective function of the linear programming model.

4.3 Number of Variables for K-MsMILP

This section gives an overview of the number of variables for K-MsMILP model. The number of variables are reported here for all four sample networks, i.e., SmallNet, NSFNET, German and European.

Topology	$\|N\|$	$\|E\|$	x_{ij}^{pq}	y_{ij}^{pq}	z_{ij}^{pq}	d_{ij}^{k}
SmallNet	10	44	9000	9000	4400	1000
NSFNET	14	42	35672	35672	8232	2744
German	16	52	78608	78608	15028	4913
European	27	108	511758	511758	78732	19683

Table 4.2: Number of variables for K-MsMILP

From the table, it is observed that the number of variables grow as the size of the topology increases. x_{ij}^{pq} and y_{ij}^{pq} grows proportional to the number of nodes in the topology. z_{ij}^{pq} increases with increasing number of direct edges in the network. d_{ij}^{k} is equal to $|N|^3$.

4.4 Comparison among Topologies

Here, we present a comparison amongst all models using different topologies based on the average performance ratio and average performance time.

4.4.1 Average Performance Ratio

The following table includes the number of nodes and number of edges in each topology for reference. It is done so the relation between the size of topology and its effect on average MLU performance can be analyzed. All the values included in this table are averaged over 20 instances. Moreover, the maximum number of segments, K, is restricted to 3.

Topology	$\|N\|$	$\|E\|$	3-sMILP	3-MsMILP
SmallNet	10	44	1	1
NSFNET	14	42	1	1
German	16	52	1.002	1
European	27	108	*	1

Table 4.3: Average Θ

It can be seen from the values of this table that the average Θ for 3-sMILP increases from the ideal value of 1 to a slightly higher value as the size of topology increases. In contrast, for 3-MsMILP, this value stays constant at 1 even as the size of topology gets bigger. For both SmallNet and NSFNET topology, the values of average Θ for both the formulations is 1. However, with the German network, 3-sMILP achieves the average Θ of 1 while 3-MsMILP was able to match the optimal results of MCF. For the European Network, 3-sMILP could not produce any results within 86400 seconds, i.e., 24 hours. In contrast, 3-MsMILP performed well by producing the average Θ of 1, on average.

4.4.2 Average Running Times

The following table gives an overview of the running time averaged over 20 instances. This is done to compare the running times of 3-MsMILP and 3-sMILP formulations.The value of maximum number of segments, K is restricted to 3 for both the models.The time noted in the following table is measured in seconds.

Topology	$\|N\|$	$\|E\|$	3-sMILP	3-MsMILP
SmallNet	10	44	2.88	1.62
NSFNET	14	42	13.93	3.54
German	16	52	289.1	22.7
European	27	108	*	2268.1

Table 4.4: Average running Time

This table demonstrates the efficiency of 3-MsMILP. It can be seen that the average running time for all the topologies has reduced greatly with this formulation. The effect has become more pronounced as the size of topology increases. For the SmallNet and NSFNET topologies, there has been a decrease in running time from 2.88 seconds to 1.62 seconds and from 13.93 seconds to 3.54 seconds, respectively. For the larger topology of German Network, with 16 nodes and 52 edges, the reduction in running time has become more pronounced. The average running time for 20 instances has reduced from 289.1 seconds to 22.7 seconds which is 92% decrease in the average running time. Moreover, for European Network, 3-sMILP was unable to produce any results within 86400 seconds, i.e., 24 hours. However, 3-MsMILP outperformed the older formulation and produced optimal results in an average of 2268.1 seconds.

4.5 Performance of Topologies

Here, we will present our results of the performance metric Θ which is the ratio of the Maximum Link Utilization of the model with MCF. The results are acquired for all four topologies for 20 network instances. For each network instance, the value of maximum number of segments, K, is limited to 3. The results for both 3-sMILP and 3-MsMILP are computed and shown here for a comparative analysis.

4.5.1 Θ for SmallNet Topology

Here, we will present the comparative graph of Θ parameter for 3-sMILP and 3-MsMILP on SmallNet topology.

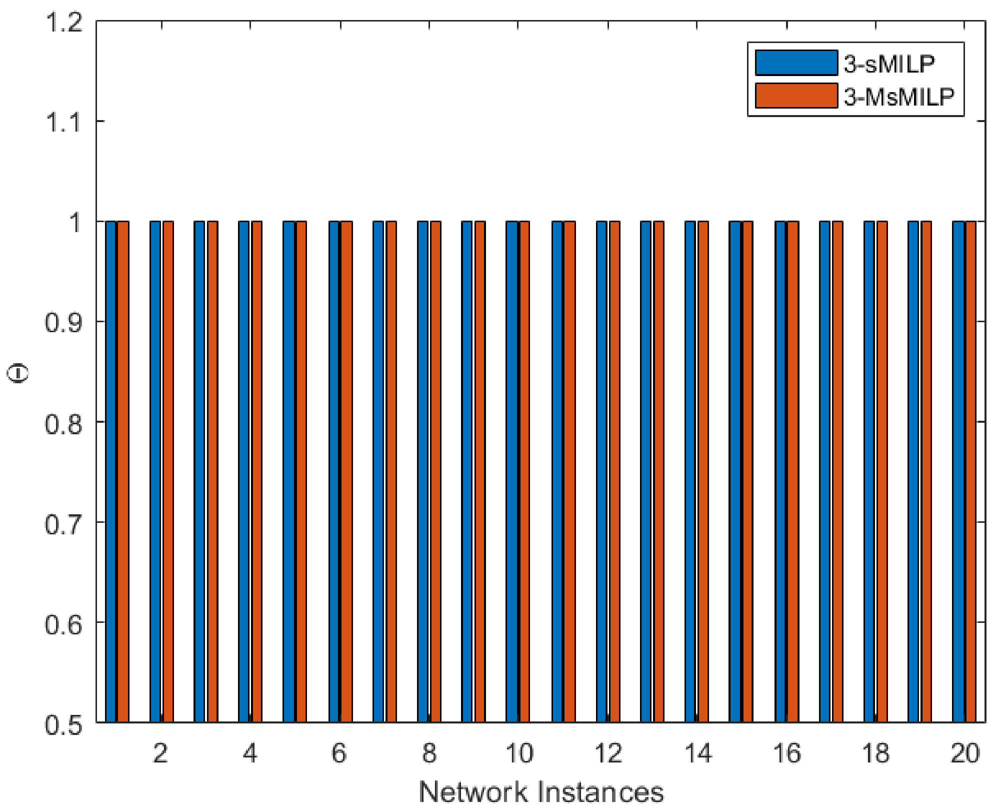

Figure 4.5: Θ for the SmallNet Topology

Figure 4.5 shows the Θ parameter for SmallNet topology for both 3-sMILP and 3-MsMILP formulations. All 20 instances are shown here. As it can be seen in the figure, the performance for both formulations is similar to the performance of MCF formulation for all network instances. It implies that the new formulation performs as good as the 3-sMILP formulation on SmallNet topology. The value 1 of Θ shows that the performance of the formulation is ideal.

4.5.2 Θ for NSFNET Topology

Here, we will present the comparative graph of Θ parameter for 3-sMILP and 3-MsMILP on NSFNET topology.

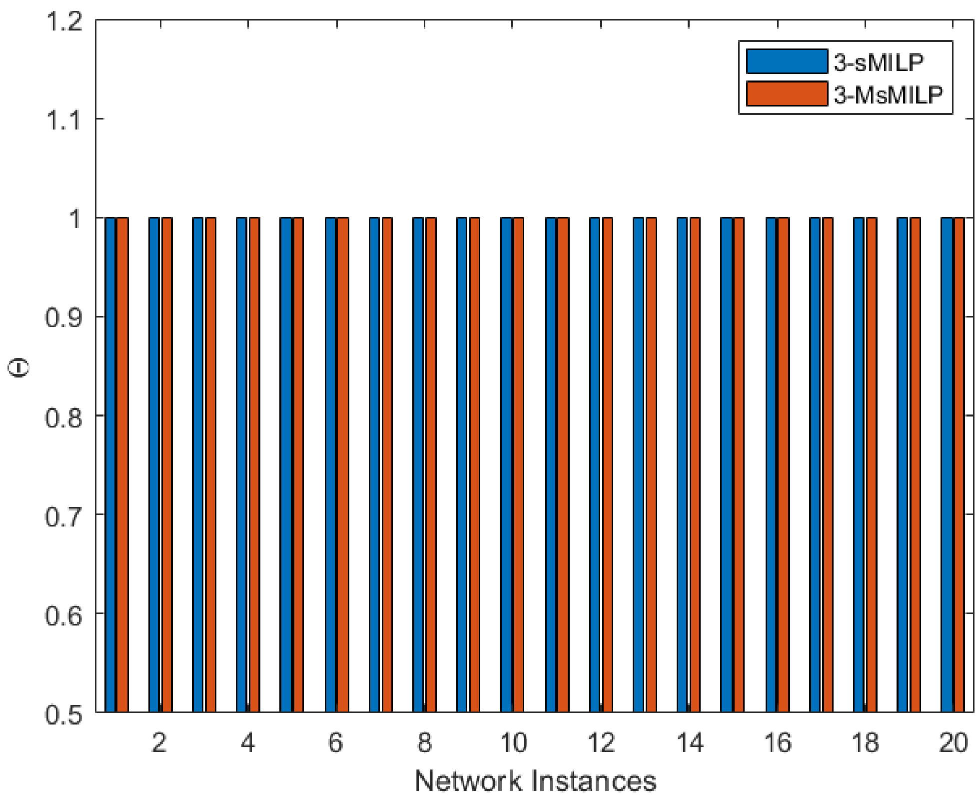

Figure 4.6: Θ for the NSFNET Topology

Figure 4.6 shows the Θ parameter for NSFNET topology for both 3-sMILP and 3-MsMILP formulations. All 20 instances are shown here. As it can be seen in the figure, the performance for both formulations is similar to the performance of MCF formulation for all network instances. It implies that the new formulation performs as good as the 3-sMILP formulation on NSFNET topology. The value of 1 of Θ shows that the performance of the formulation is ideal.

4.5.3 Θ for German Topology

Here, we will present the comparative graph of Θ parameter for 3-sMILP and 3-MsMILP on German topology.

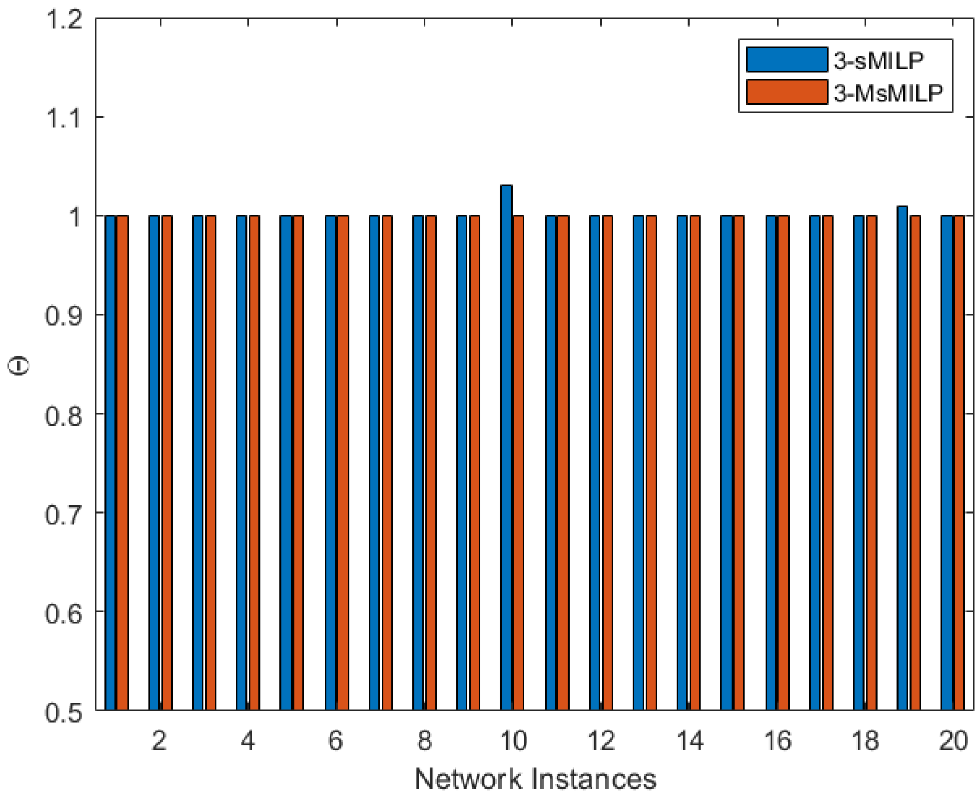

Figure 4.7: Θ for the German Topology

Figure 4.7 shows the Θ parameter for German topology for both 3-sMILP and 3-MsMILP formulations. All 20 instances are shown here. As it can be seen in the figure, the performance ratio parameter, Θ, of 3-MsMILP always equals 1 for all 20 network instances. For 3-sMILP, the performance has not been ideal always. For example, in network instance 10, the value of Θ for 3-sMILP exceeds 1 but it remains 1 for our formulation. Similarly, for network instance 18, Θ slightly exceeds 1 for the 3-sMILP formulation where our formulation 3-MsMILP gave ideal Θ. Thus, our formulation gives ideal performance even when the size of network increases.

4.5.4 Θ for European Topology

Here, we will present the graph of Θ parameter for 3-MsMILP model on European topology.

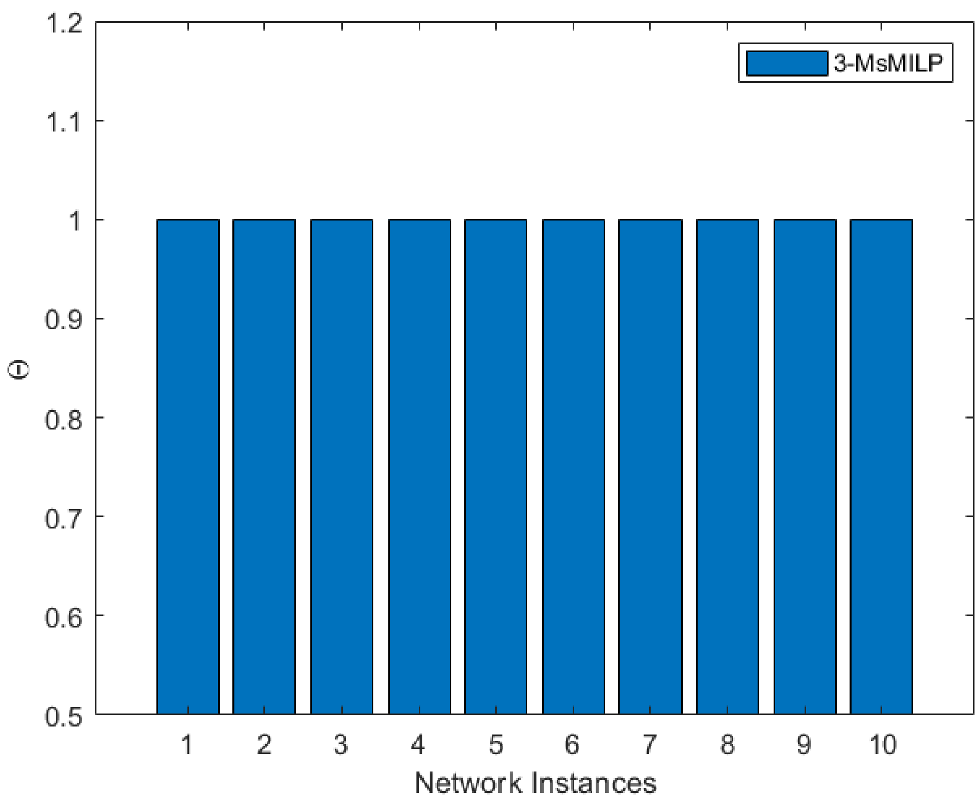

Figure 4.8: Θ for the European Topology

Figure 4.8 shows the Θ parameter for European topology for 3-MsMILP model. For 3-sMILP, the model could not yield any results, either optimal or sub-optimal, within a specified time. Hence, the results for 3-sMILP are not included here. The results are obtained for 10 network instances. From the figure, it can be seen that the 3-MsMILP formulation gives ideal performance for all network instances. Hence, our formulation produces the performance ratio of 1 for even a larger topology.

4.6 Running Times for Different Topologies

Here, we will present our results of the running times of both models. All the time values reported here are in seconds. The results are acquired on three topologies for 20 network instances. For European network, results are obtained on 10 network instances. For each network instance, the value of maximum number of segments, K, is limited to 3. The results for MCF, 3-sMILP and 3-MsMILP are computed and shown here for a comparative analysis.

4.6.1 Running Times for SmallNet Topology

Here, we will present the comparative graph of running time for MCF, 3-sMILP and 3-MsMILP on SmallNet topology.

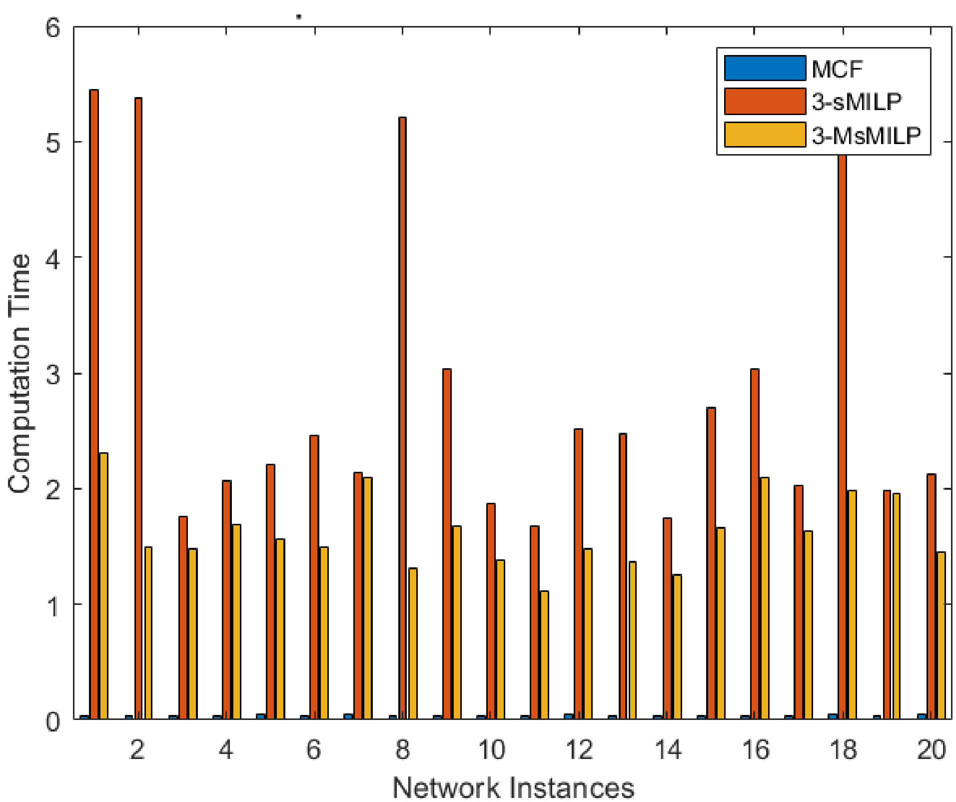

Figure 4.9: Running Times for the SmallNet Network

Figure 4.9 shows the running times for MCF, 3-sMILP and 3-MsMILP. For the SmallNet topology, the running time with MCF was very small. It has remained less than 0.1 seconds for all 20 instances. In contrast, 3-sMILP produces a significantly large running time as compared to MCF with the highest value approaching 5.44 seconds for network instance 1. With 3-MsMILP, the running time is close to the running time of MCF. For example, for network instance 2, 3-sMILP produces optimal results in 5.38 seconds and 3-MsMILP produces optimal results in 1.5 seconds. The reduction in running time has been observed for all network instances where our new formulation reduced this time and drove it closer to the running time of MCF.

4.6.2 Running Times for NSFNET Topology

Here, we will present the comparative graph of running time for MCF, 3-sMILP and 3-MsMILP on the NSFNET topology.

Figure 4.10 shows the running times for MCF, 3-sMILP and 3-MsMILP on NSFNET topology. The running time with MCF was the smallest here as well and it remained less than 0.2 seconds for all 20 instances. 3-sMILP produces a very large running time with the highest value of 37.85 seconds for network instance 18. Our new formulation, 3-MsMILP, significantly reduced the running time as compared to 3-sMILP. For network instance 18, 3-sMILP took 37.85 seconds to produce optimal results but with 3-MsMILP, this time was reduced to 3.93 seconds which is very close to the time taken by MCF. Similarly, for all other instances on NSFNET topology, the running time for 3-MsMILP was closer to the running time of MCF.

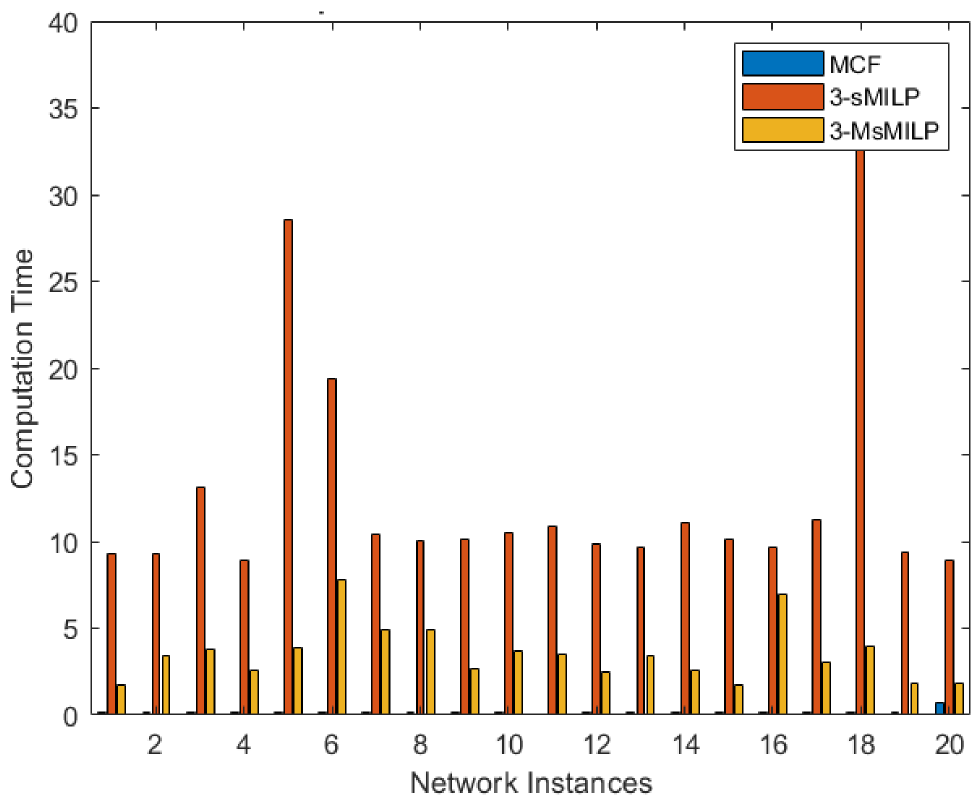

Figure 4.10: Running Times for the NSFNET Network

4.6.3 Running Times for German Topology

Here, we will present the comparative graph of running time for MCF, 3-sMILP and 3-MsMILP on the German topology.

Figure 4.11 shows the running times for three models of MCF, 3-sMILP and 3-MsMILP on German topology. On this network, MCF model produces optimal results in less than 0.2 seconds for all 20 network instances. As the size of this network is very large, 3-sMILP took a very long time to produce optimal results. The largest time is observed for network instance 2 where 3-sMILP took 1376.95 seconds to give results. However, for this instance, 3-MsMILP produces results in 16.60 seconds which is significantly less than the time compared to 3-sMILP. This pattern has been observed for all 20 instances. For 3-MsMILP, the running time remained less than 50 seconds which is a significant reduction from the high running time produced by 3-sMILP.

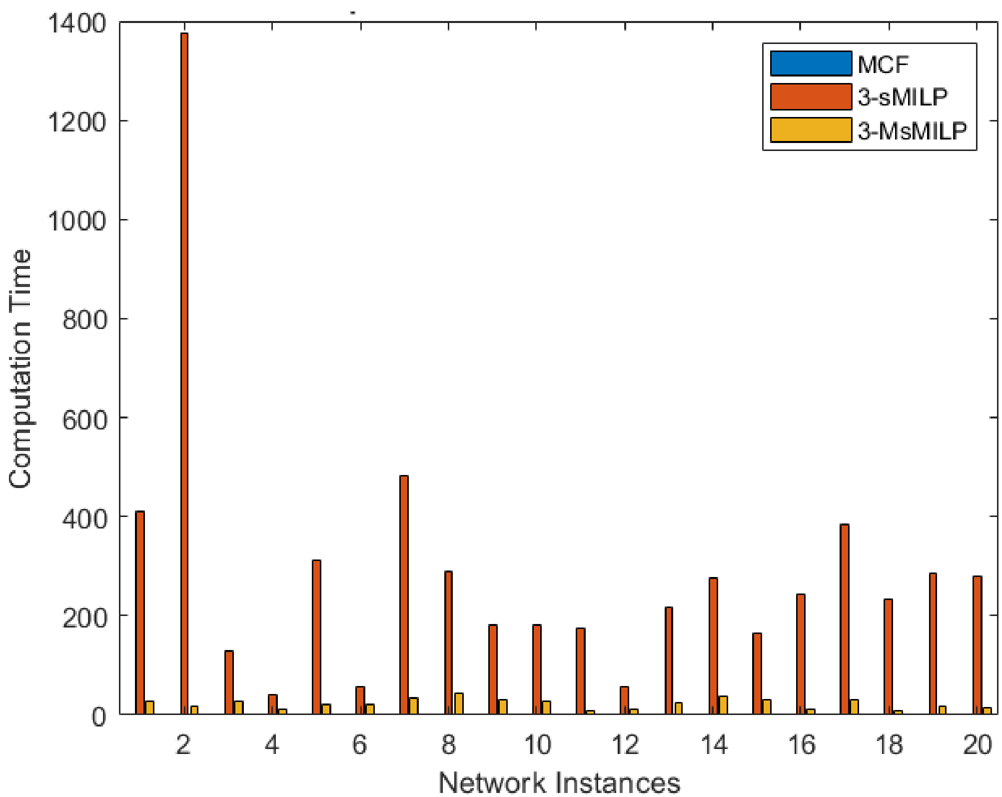

Figure 4.11: Running Times for the German Network

4.6.4 Running Times for European Topology

Here, we will present the comparative graph of running time for MCF and 3-MsMILP on the European topology.

Figure 4.12 shows the running times for 3-MsMILP and MCF models on European topology for 10 network instances. Here, the results for 3-sMILP are not included as no optimal results were obtained for any of the instances within a specified time. Also, as the running time on the European topology is very long, results are obtained only for 10 network instances. As it can be seen from the figure, 3-MsMILP produces optimal results for all network instances within a considerable period of time. MCF model on European topology produces results in less than 5 seconds for all network instances. 3-MsMILP produces optimal results in less than 4000 seconds for all instances. The least time taken by 3-MsMILP is 547.44 seconds for network instance 6 where MCF produces the results for same

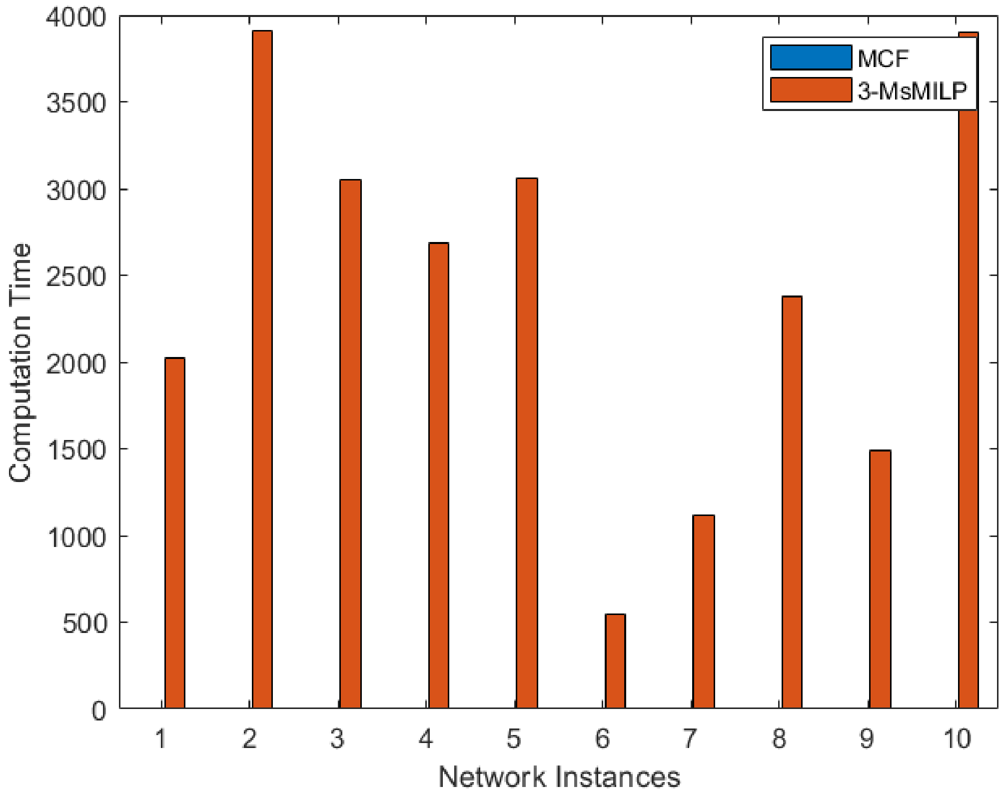

Figure 4.12: Running Times for the European Network

instance in 2.49 seconds. Overall, as compared to 3-sMILP, our new formulation performs more efficiently.

4.7 Performance of K-MsMILP for different values of K

This section covers the results of our new formulation K-MsMILP for different values of K, i.e., the maximum number of segments in a path. These results are obtained using all four topologies for 10 network instances. The values of K are varied and the effect of this variation is analyzed on the performance metric Θ.

4.7.1 Θ for SmallNet Topology for different K-values

Here, we will present the results of MLU for SmallNet network for three values of K, i.e., $K = 3$, $K = 2$ and $K = 1$. The results are shown in Table 4.5.

Instance	3-MsMILP	2-MsMILP	1-MsMILP
1	1	1	2.39
2	1	1	2.77
3	1	1	2.73
4	1	1	2.48
5	1	1	2.83
6	1	1	4.08
7	1	1	3.28
8	1	1	4.21
9	1	1	3.82
10	1	1	3.33

Table 4.5: Θ for the SmallNet Network for different K-values

The table shows that the values of Θ for SmallNet topology for $K = 3$ and $K = 2$ always remain 1. For $K = 1$, the performance metric shows a high variation from the lowest value of 2.39 for network instance 1 and the highest value of 4.21 for network 8. Overall, it can be observed from these results that the values of Θ will diverge from the ideal behavior if the value of K is lowered.

4.7.2 Θ for NSFNET Topology for different K-values

Here, we will present the results of MLU for NSFNET network for three values of K, i.e., $K = 3$, $K = 2$ and $K = 1$. The results are shown in Table 4.6.

Similar to the behaviour shown by SmallNet topology, NSFNET topology also yields $\Theta = 1$ for $K = 3$ and $K = 2$ for all network instances. For $K = 1$, the value of Θ always shows non-ideal values with the highest value of 2.41 for network instance 8 and the lowest value of 1.45 for network instance 2.

Instance	3-MsMILP	2-MsMILP	1-MsMILP
1	1	1	1.72
2	1	1	1.45
3	1	1	2.18
4	1	1	1.63
5	1	1	1.94
6	1	1	1.83
7	1	1	2.07
8	1	1	2.41
9	1	1	1.76
10	1	1	2.38

Table 4.6: Θ for the NSFNET Network for different K-values

4.7.3 Θ for German Topology for different K-values

Here, we will present the results of MLU for German network for three values of K, i.e., $K = 3$, $K = 2$ and $K = 1$. The results are shown in Table 4.7.

Instance	3-MsMILP	2-MsMILP	1-MsMILP
1	1	1	1.45
2	1	1	1.96
3	1	1.01	2.72
4	1	1	2
5	1	1	1.99
6	1	1.02	1.88
7	1	1	2.45
8	1	1	2.20
9	1	1	2.21
10	1	1	1.95

Table 4.7: Θ for the German Network for different K-values

Table 4.6 shows the values of Θ for German Network. For $K = 3$, the value of Θ is always 1 for all network instances. For $K = 2$, table shows some instances where the performance of K-MsMILP is deviant from the ideal MCF values. For network instance 3, the value for $K = 2$ is 1.01. It is 1.02 for network instance 6. For $K = 1$, the values always remain higher than 1. The highest value of Θ for $K = 1$ is observed at network instance 3 which is 2.72. The lowest value is 1.45 for network instance 1.

4.7.4 Θ for European Topology for different K-values

Here, we will present the results of MLU for European network for three values of K, i.e., $K = 3$, $K = 2$ and $K = 1$. The results are shown in Table 4.8.

Instance	3-MsMILP	2-MsMILP	1-MsMILP
1	1	1	3.54
2	1	1	2.94
3	1	1	2.97
4	1	1	2.99
5	1	1	2.66
6	1	1	3.29
7	1	1	3.80
8	1	1	3.75
9	1	1	4.03
10	1	1	2.26

Table 4.8: Θ for the European Network for different K-values

The table shows ideal performance of K-MsMILP on European Network for $K = 3$ and $K = 2$. For all network instances, the value remains 1 which is the ideal performance metric. For $K = 1$, the performance metric shows non-ideal values for all instances. The highest value of Θ is 4.03 for network instance 9. The lowest value of Θ for $K = 1$ is 2.26 for network instance 10. It is clearly observed that the Θ increases as the value of K is decreased from a higher value to 1 for all topologies.

4.8 Performance with Length Constraint

This section covers the results of our new model, K-MsMILP with length constraint. The performance metric Θ is reported and analyzed here when there is an additional constraint in the model which restricts the maximum path length traversed by the traffic. Here, the results are obtained for 20 network instances by restricting the maximum number of segments, K, to 3. The results include three parameters where the value of length parameter, α is changed from 2 to 1.

Here, $\alpha = 2$ implies that the maximum path length for the traffic can be twice the shortest path length. For $\alpha = 1$, the maximum path length traversed by the traffic equals the shortest path length.

4.8.1 Θ with Length Constraint for SmallNet Topology

Here, we will present the comparative graph of Θ with length constraint for 3-MsMILP on SmallNet topology.

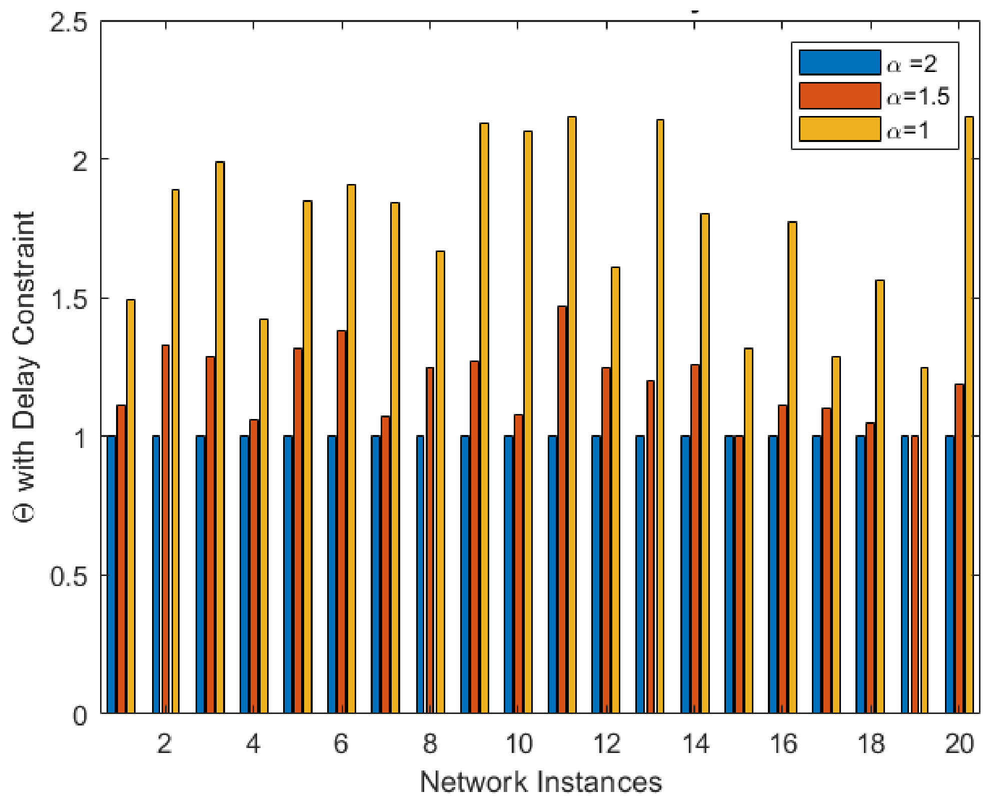

Figure 4.13: Θ with Length Constraint for the SmallNet Network

Figure 4.13 shows Θ with length constraint for 3-MsMILP on SmallNet topology. For all 20 instances, the value of the performance metric Θ has shown a deviation from an ideal value of 1 as the value of α is reduced from a higher value to the value of 1. Moreover, a lower value of Θ is observed for the higher value of α. It shows that for SmallNet topology, traffic traverses paths which have longer path lengths. In other words, traffic utilizes direct links to reach the destination. For all instances, with $\alpha = 2$ the value of Θ always remains 1. For network instance 11, $\alpha = 1.5$ gives $\Theta = 1.47$. Similarly, with $\alpha = 2$, the value of Θ reaches 2.15. This is a high aberration from the ideal value of 1 for Θ which is observed as the α is reduced.

4.8.2 Θ with Length Constraint for NSFNET Topology

Here, we will present the comparative graph of Θ with length constraint for 3-MsMILP on NSFNET topology.

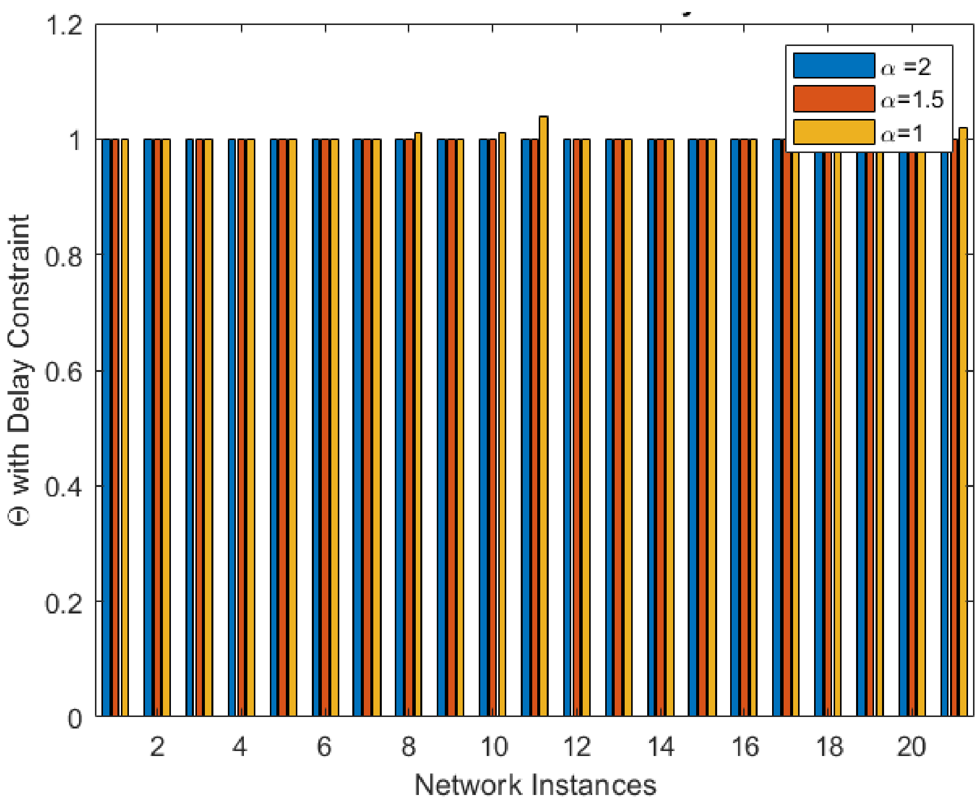

Figure 4.14: Θ with Length Constraint for the NSFNET Network

Figure 4.14 illustrates the performance metric Θ for NSFNET topology when the length constraint is introduced in the formulation. Here, for all instances, the value of α is changed from the value of 2 to the value of 1. When the value of α equals 2, the corresponding value of Θ is always 1 for all network instances. For NSFNET topology, $\alpha = 1.5$ also yields an ideal value of Θ. However, for $\alpha = 1$, some network instances show a deviation from the ideal value. For network instance 11, the value of Θ is 1.04 which is higher than the ideal performance value.

As compared to SmallNet topology, NSFNET topology shows fewer instances where the performance of 3-MsMILP is affected as the value of α is reduced. It shows that SmallNet has more flows which route through the longer path lengths when there is a lose constraint on α. For NSFNET network, paths with longer lengths are limited and the flow usually follows the shortest length paths.

4.8.3 Θ with Length Constraint for German Topology

Here, we will present the comparative graph of Θ with length constraint for 3-MsMILP on German topology.

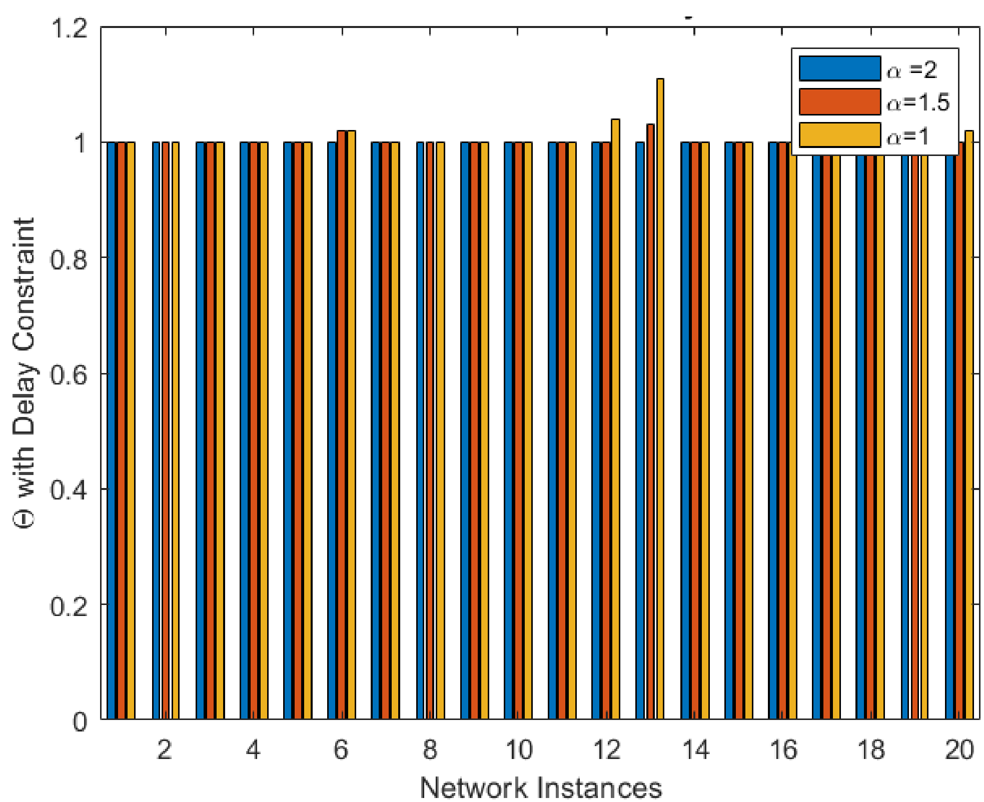

Figure 4.15: Θ with Length Constraint for the German Network

Figure 4.15 shows 20 network instances and the values of Θ when the length constraint is included in the formulation. The results are obtained for three values of α as well. The relationship between α and Θ remains similar over here as well. As the value of α increases, the value of Θ becomes more ideal. For German network, an ideal performance metric Θ is obtained for all instances when $\alpha = 2$. As the value of α decreases to 1.5, some instances show a deviation from the ideal performance. Decreasing α further continues this deviation. For network instance 13, Θ increases to 1.03 for $\alpha = 1.5$, It increases more to the value of 1.11 as $\alpha = 1$. Hence, for German topology, some network instances utilize the direct links and traverse the traffic through relatively longer paths to optimize the performance metric Θ.

Similar to the behavior of NSFNET topology as compared to SmallNet topology, German network also shows fewer instances of variation where decreasing the value of α affects the performance metric Θ. Therefore, as compared to SmallNet network, for German network, there exists fewer alternative paths for the flow to be routed. The flow usually follows the shortest length path. However, as compared to the NSFNET network, there exist more instances of variation of Θ in German network with length constraint. The aberrant values of Θ are more frequent as well as higher. This shows that the flow has more options to follow in German network as compared to the NSFNET network. These options are reduced as the value of α is brought closer to the shortest path length.

4.8.4 Θ with Length Constraint for European Topology

Here, we will present the comparative graph of Θ with length constraint for 3-MsMILP on European topology.

Figure 4.16 shows the relationship between α and Θ for 10 network instances on European topology. For $\alpha = 2$, Θ always shows an ideal performance. As the value of α decreases, the performance ratio also declines. For network instance 9, decreasing the value of α deteriorates the performance ratio Θ. For $\alpha = 1.5$, Θ increases to the value of 1.05. Similarly, Θ increases to 1.07 when the value

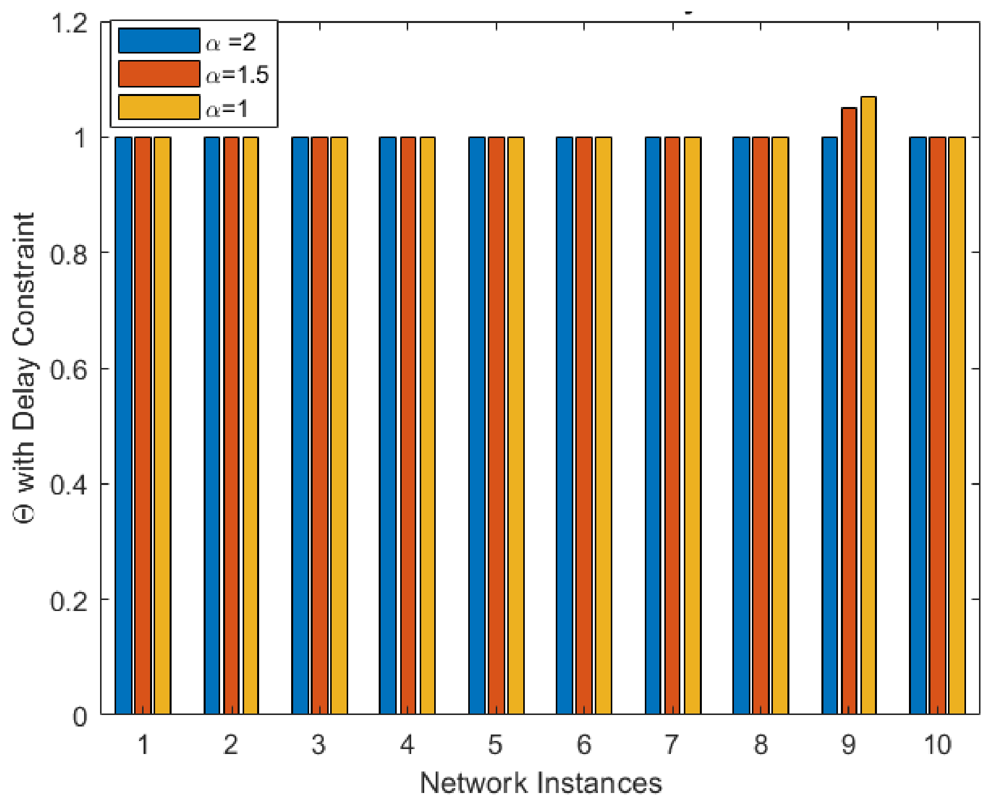

Figure 4.16: Θ with Length Constraint for the European Network

of α is reduced to 1. This signifies the importance of direct link segments in our original formulation to achieve the optimum Maximum Link Utilization.

The values of Θ for European network remain ideal for all network instances except 1. Based on this observation, it can be said that there exist even fewer alternative paths in this network for the flow to follow as compared to the other topologies. The flow is usually routed through the shortest path in the European network. Therefore, decreasing the value of α does not affect the performance of K-MsMILP to a higher degree.

4.9 Running Times with Length Constraint

This section covers the results of the new formulation, K-MsMILP with length constraint. The performance is tested based on the running times of the formulation to reach the optimum results when the maximum path length of the traffic is restricted to the smallest path length. The maximum number of segments, K, is restricted to 3 for these results. The results are obtained for 20 different network instances for three topologies i.e. SmallNet, NSFNET and German topology. For European topology, results are obtained on 10 network instances. All the results are obtained for three different values of congestion parameter, α. Initially, α is selected to be 2. It is then reduced to 1.5 and then the formulation is tweaked to ensure that the maximum path length traversed by the traffic equals the shortest path length where $\alpha = 1$.

4.9.1 Running Times with Length Constraint for SmallNet Topology

Here, we will present the comparative graph of running times with length constraint for 3-MsMILP on SmallNet topology.

Figure 4.17 shows the running times with length constraint for SmallNet topology. The results are obtained for 3 values of α. In all of the instances, the maximum running time was limited to 10800 seconds to obtain optimal results. Here, we have observed that the running time and the path length traversed by the traffic do not share any patterned relationship. For changing values of α, the running time may increase or decrease for different instances. For network instance 10, $\alpha = 2$ produces optimal results 158.86 seconds. For $\alpha = 1.5$, optimal results were obtained in 10800 seconds and for $\alpha = 1$, the model produced optimal results in 27.93 seconds. In contrast, for network instance 15, $\alpha = 2$ took the least amount of time to produce results which is 8.7 seconds. Then, $\alpha = 1.5$ took 20.25 seconds and $\alpha = 1$ produced optimal results in 10800 seconds.

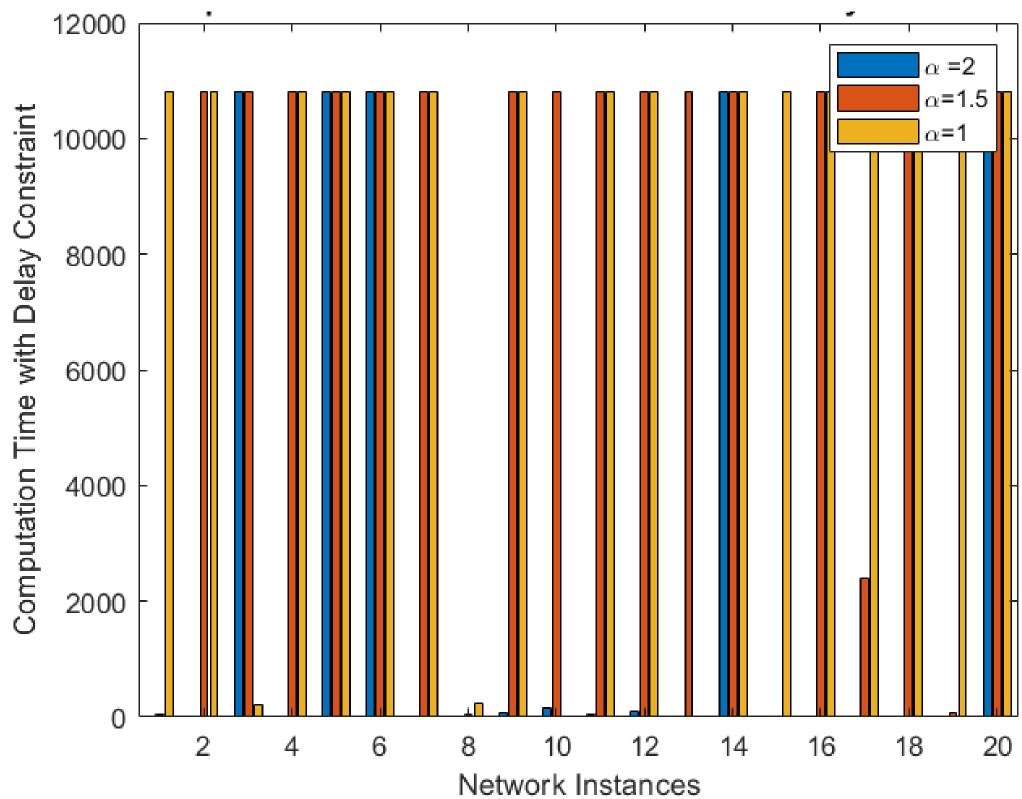

Figure 4.17: Running Times with Length Constraint for the SmallNet Network

4.9.2 Running Times with Length Constraint for NSFNET Topology

Here, we will present the comparative graph of running times with length constraint for 3-MsMILP on NSFNET topology.

Figure 4.18 shows the running times with length constraint for NSFNET topology. For all 3 values of α, the running time is noted for all 20 instances. For this topology, again, there seemed to be no relationship between α and running time. For network instance 1, $\alpha = 2$ produced optimal results in 56.62 seconds. For the same instance, $\alpha = 1.5$ took 24.53 seconds and $\alpha = 1$ took 36.64 seconds. However, for network instance 10, decreasing value of α increased the running time of the model. For this instance, it took 41.72 seconds, 61.97 seconds and 850.20 seconds to produce optimal results for $\alpha = 2$, $\alpha = 1.5$ and $\alpha = 1$ respectively.

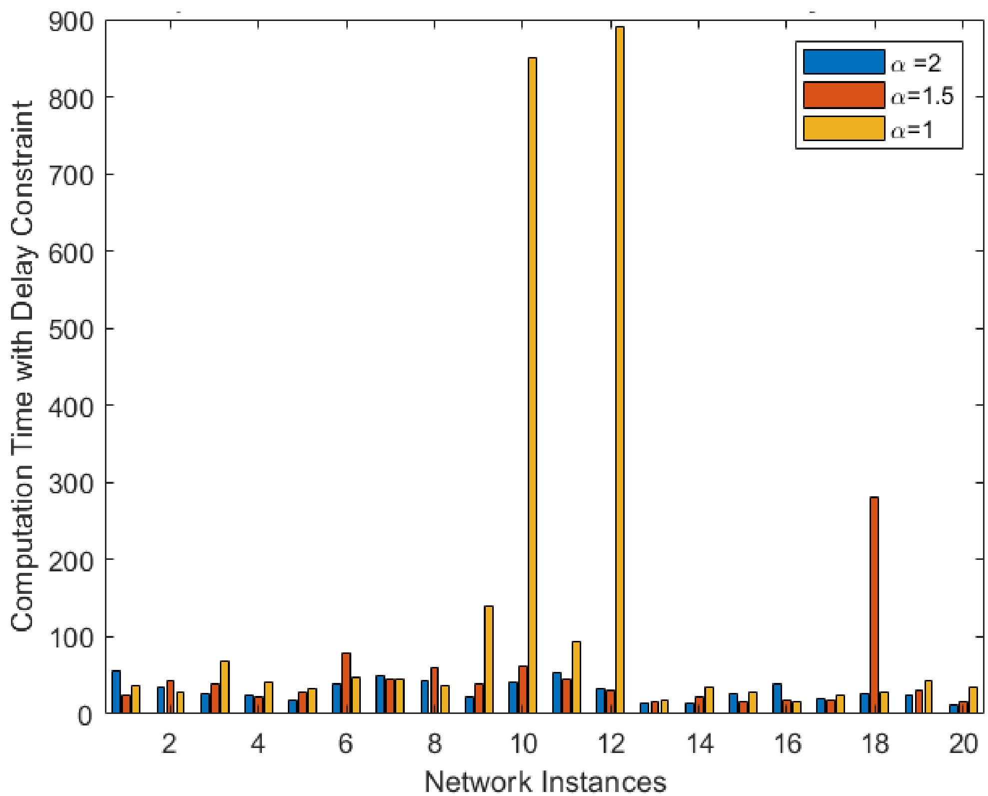

Figure 4.18: Running Times with Length Constraint for the NSFNET Network

4.9.3 Running Times with Length Constraint for German Topology

Here, we will present the comparative graph of running times with length constraint for 3-MsMILP on German topology.

Figure 4.19 shows the running times with length constraint on German topology. For all 20 instances shown in the graph, it is difficult to infer a relationship between α and the corresponding running time. For German network, the running time may increase or decrease as the value of the congestion factor is decreased. For network instance 9, it increases as the value of α is decreased. For $\alpha = 2$, the running time is 553.36 seconds. It increases to 693.67 seconds for $\alpha = 1.5$ and increases further to 894.10 seconds when the value of congestion factor is 1. For network instance 11, it shows a very different pattern. For $\alpha = 2$, it gives 441.43 seconds. The highest running time for this instance is observed for $\alpha = 1.5$ when

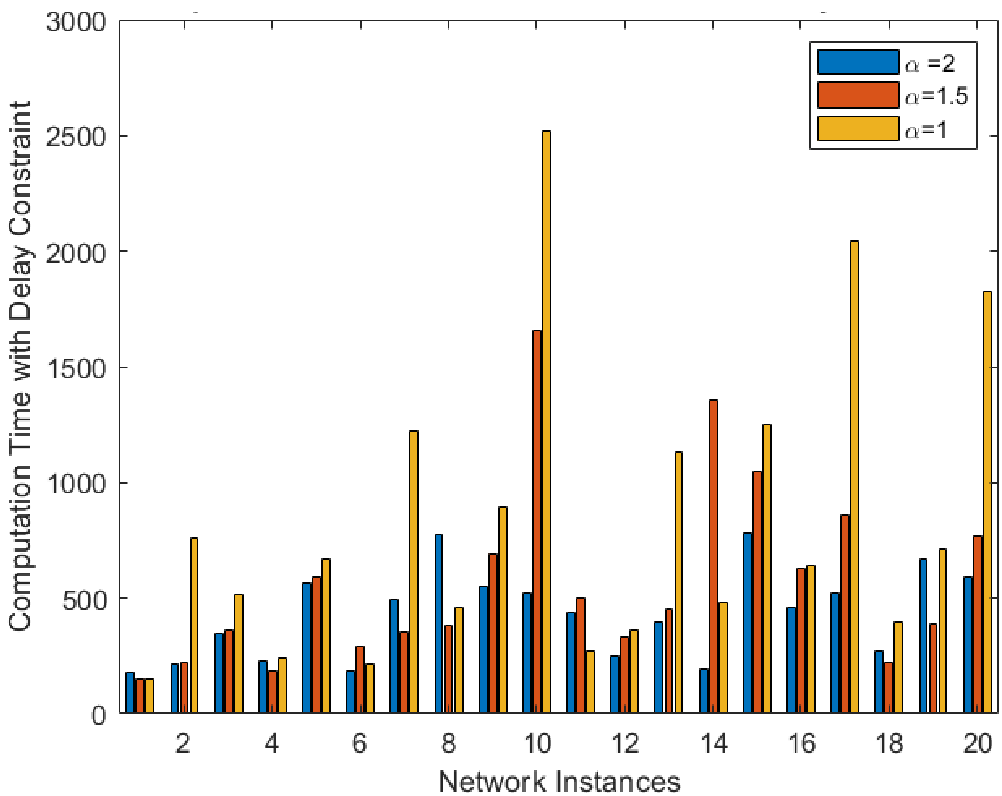

Figure 4.19: Running Times with Length Constraint for the German Network

the running time reaches 500.83 seconds. The running time drops back to 266.80 seconds for $\alpha = 1$.

4.9.4 Running Times with Length Constraint for European Topology

Here, we will present the comparative graph of running times with length constraint for 3-MsMILP on European topology.

The above figure gives the running times with length constraint on European network. The results are obtained on 10 network instances by changing the values of congestion factor for each instance and obtaining the results. To save time, the maximum limit on running time to produce optimal results was 86400 seconds for all network instances. For this topology, changing the values of congestion

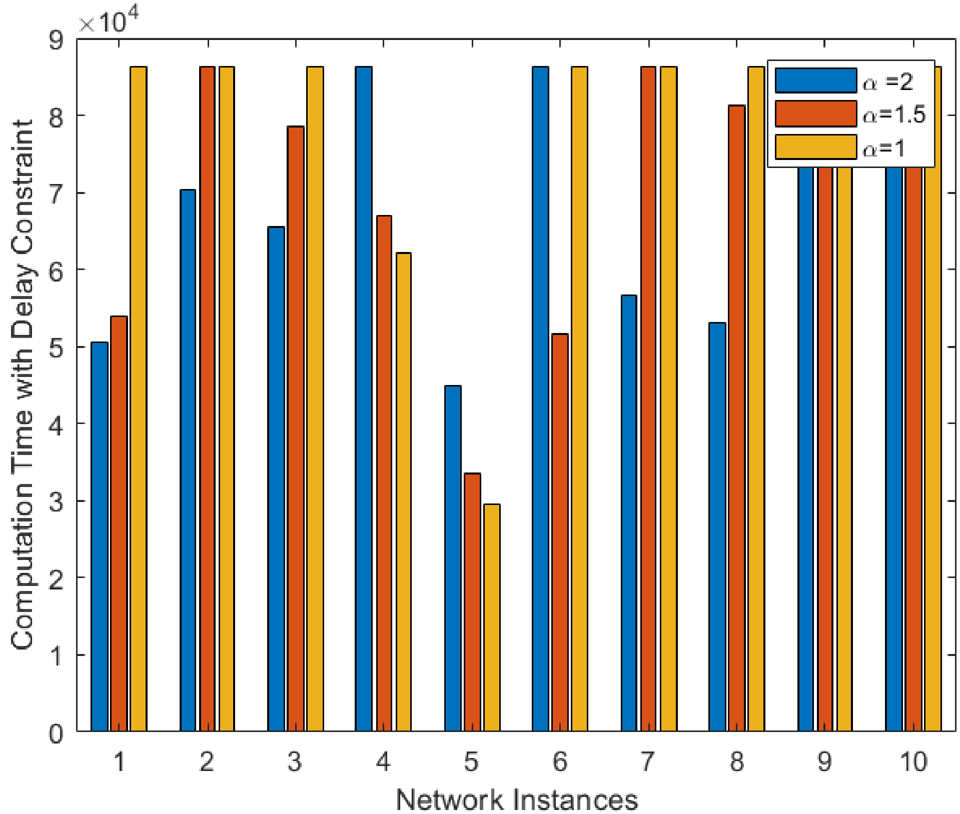

Figure 4.20: Running Times with Length Constraint for the European Network

factor do not impact the running time in a regular pattern. For network instance 1, the running time increases as the value of α is reduced. On the other hand, network instance 5 shows a decrease in running time with decreased values of α.

4.10 Average Running Times for 3-MsMILP and 3-MsMILP with Length Constraint

Here, we present the average running times for 3-MsMILP and 3-MsMILP with length constraint. The results are reported here for four topologies, i.e., Small-Net, NSFNET, German and European. In these results $\alpha = \infty$ is the average running time for 3-MsMILP without any additional constraints. These results are presented to show the increase in the running times of the formulations as length constraints are added.

4.10.1 Average Running Times for 3-MsMILP and 3-MsMILP with Length Constraint for SmallNet Topology

Here, we present the average running times for 3-MsMILP formulation with and without length constraint using SmallNet topology.

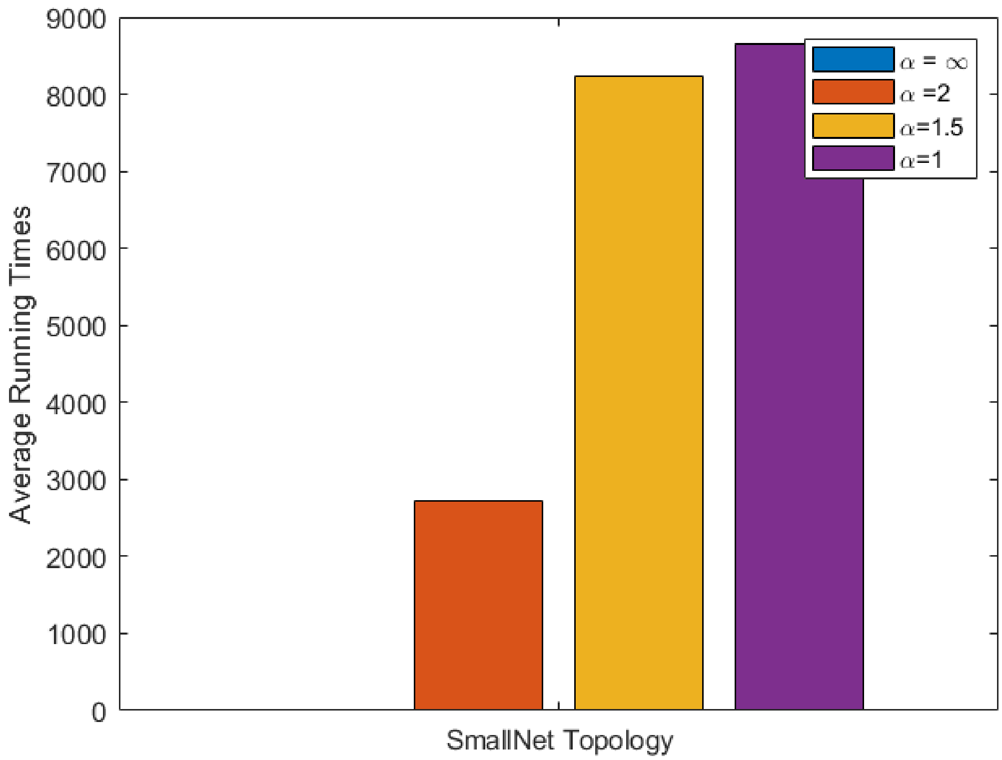

Figure 4.21: Average Running Times with Length Constraint for the SmallNet Network

On average, the running times increase significantly with the length constraint. As the value of length factor, α is reduced, the running times increase further. There remains a significant difference between the running times where $\alpha = \infty$

and where the value of α is lowered.

4.10.2 Average Running Times for 3-MsMILP and 3-MsMILP with Length Constraint for NSFNET Topology

Here, we present the average running times for 3-MsMILP formulation with and without length constraint using NSFNET topology.

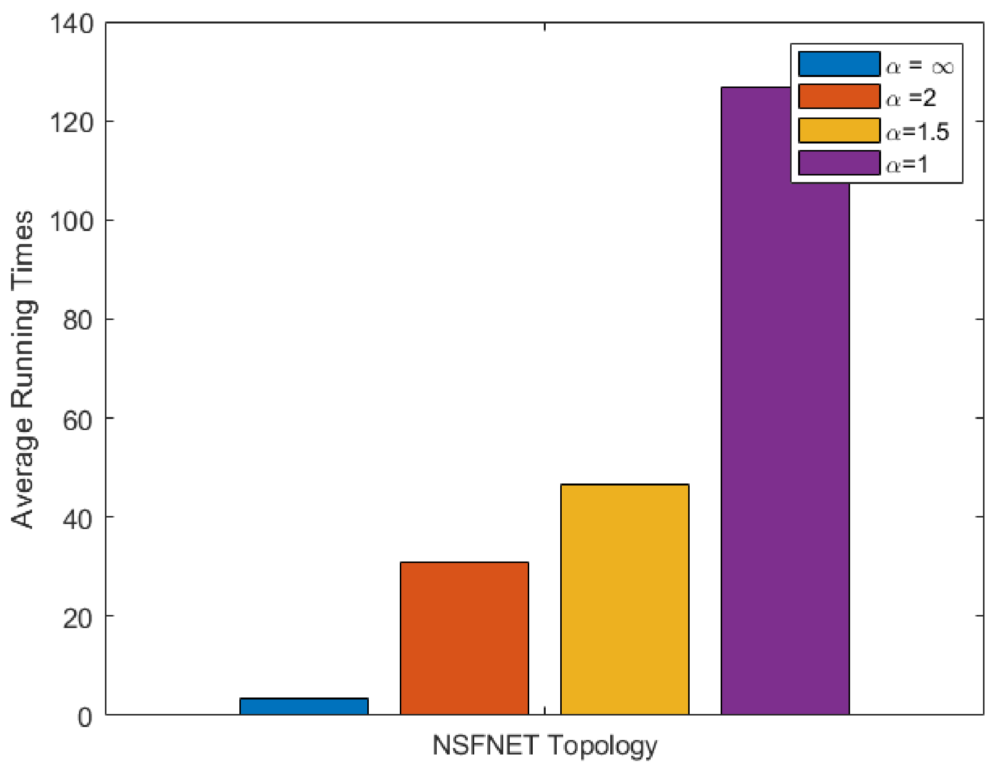

Figure 4.22: Average Running Times with Length Constraint for the NSFNET Network

For NSFNET topology, the average times with the addition of length constraint increases as well. This increase becomes more prominent as the value of α is reduced.

4.10.3 Average Running Times for 3-MsMILP and 3-MsMILP with Length Constraint for German Topology

Here, we present the average running times for 3-MsMILP formulation with and without length constraint using German topology.

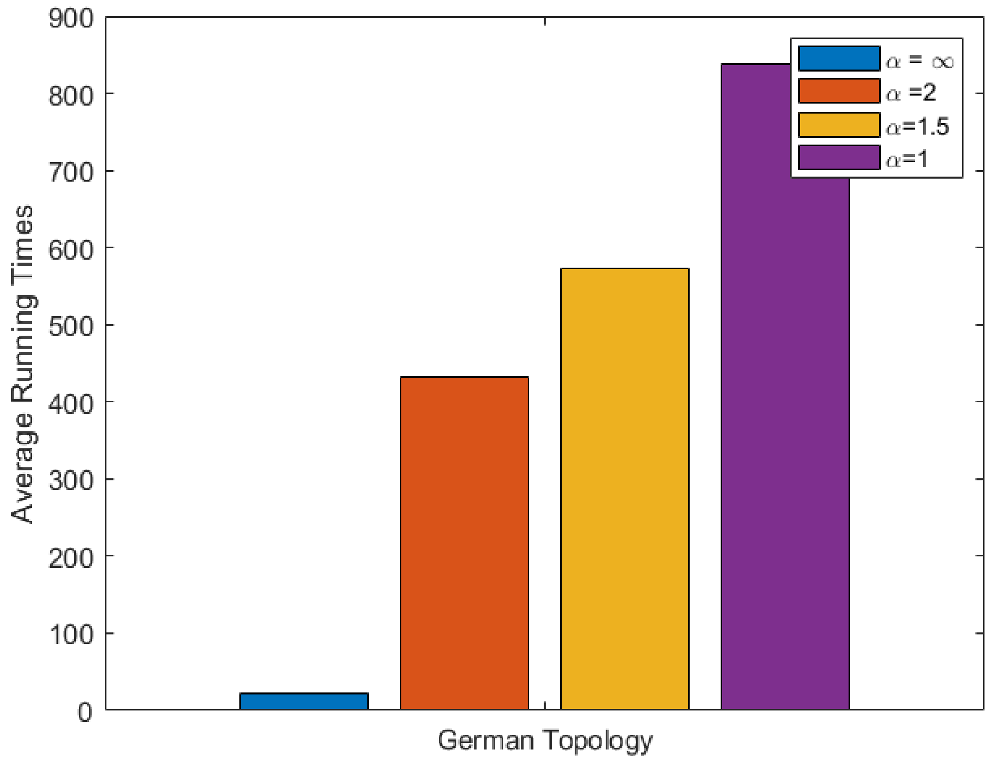

Figure 4.23: Average Running Times with Length Constraint for the German Network

The average running times for German network using 3-MsMILP and 3-MsMILP model with length constraint are significantly different. Addition of length constraint increases the running times. This time increases more as the value of length factor α is reduced.

4.10.4 Average Running Times for 3-MsMILP and 3-MsMILP with Length Constraint for European Topology

Here, we present the average running times for 3-MsMILP formulation with and without length constraint using European topology.

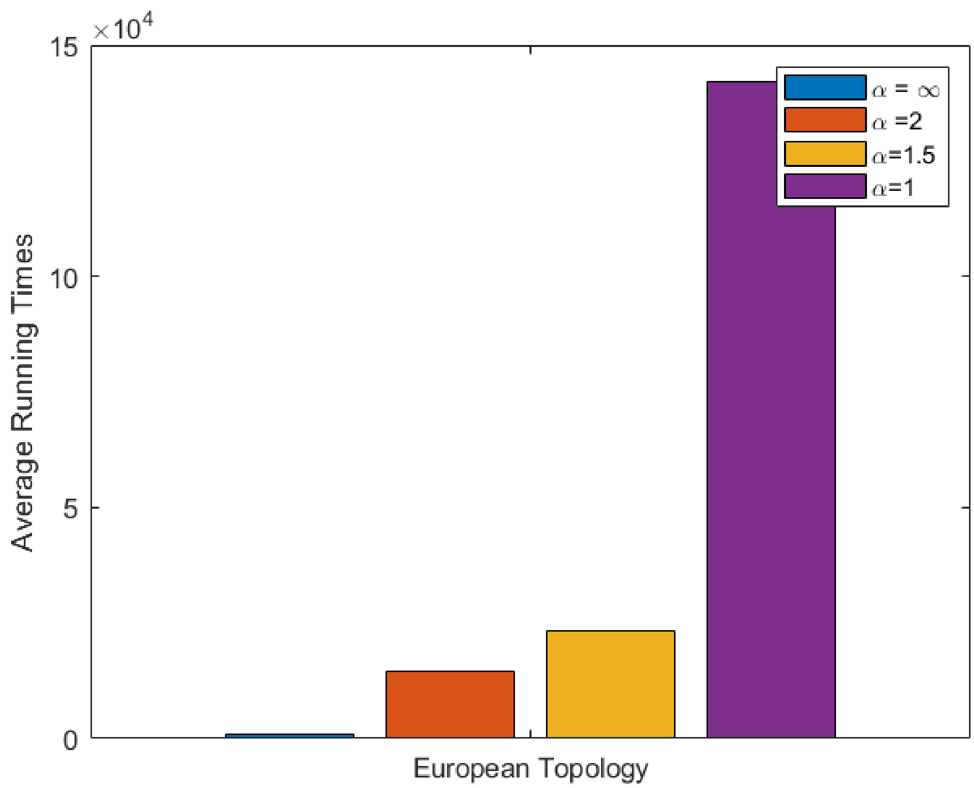

Figure 4.24: Average Running Times with Length Constraint for the European Network

There remains a significant difference in the running times of 3-MsMILP and 3-MsMILP with length constraint for European network as well. Addition of length constraint elongates the running times and reducing the value of α increases it further.

Chapter 5

Conclusion

Segment routing is an important and relatively new technique for traffic engineering in the Internet. With its many advantages, segment routing remains a relatively unexplored paradigm in the routing problems. Previous studies have explored the Integer Linear Programming (ILP) and Mixed Integer Linear Programming (MILP) models to route the traffic flows in the network with the objective of minimizing the Maximum Link Utilization (MLU) of the network. These models weighed the efficiency and efficacy of their models against the performance showed by the classic Multi-Commodity Flow (MCF) formulation. There remain certain short-comings of these models which include a large number of variables and constraints which either increased the MLU to a higher value or increased the running time of these models. For larger topologies, the running time always remained high for these formulations. Moreover, the models did not fully incorporate all the segment-pairs present in the network.

We have introduced two new models, K-MMILP and K-MsMILP, which utilize all the segment-pairs and the direct links in the network for traffic flows. Inspired by the previous works, K-MMILP uses path-based approach to route the traffic whereas K-MsMILP utilizes flow-based approach to form valid paths for the traffic to follow. These models have a fewer number of variables and constraints with the same objective of minimizing the Maximum Link Utilization (MLU) of the network. The new models are designed to be more efficient in terms of load balancing and running times. As our formulations do not always follow the shortest-path to route the traffic, we have also studied the effect of limiting the maximum path length of the flow by introducing a length constraint in our original simplified formulation.

K-MMILP is an extensive approach which uses path-based approach and is time-consuming to solve. So, we have obtained results for K-MsMILP on four different topologies using Gurobi Optimization Solver. We have observed that our model always produces ideal MLU values for all network sizes. The ideal value of MLU for all networks is derived from the corresponding classical MCF formulation. It has also been observed that K-MsMILP always reduced the running times of the solutions for all the topologies bringing them closer to the running times of MCF. The reduction in the running time of the formulations became more pronounced as the size of topology increased. While the K-sMILP was unable to yield optimal solutions for larger topologies such as the European network, K-MsMILP was able to produce optimal solutions while achieving relatively small running times. For instance, for German Network the running time is reduced by a factor of $82.9 - 2.85$ times with an average of 14.9.

While observing the results of the K-MsMILP with length constraint, it was noted that decreasing the maximum path length traversed by the traffic did increase the MLU of the network for some instances. This behaviour was most ardently observed in SmallNet topology demonstrating that there exist more alternative paths for the flow to follow in this network. For all other topologies, the effect of reducing the length factor, α, was either not very pronounced or infrequent showing that the flow follows the shortest path in these topologies usually. Moreover, the running time showed a variation for decreasing the length factor, α. However, this variation did not follow a regular pattern for all topologies. It was observed that the running times increased significantly for SmallNet topology when the length factor was reduced given the more number of available alternative paths in this topology for the flow to follow. For other three topologies, as the flow was mostly routed through the shortest paths, the variation was not as significant.

Future research can be done to explore the effects of choosing the network cost from a distribution other than uniform distribution such as tailed distribution or exponential distribution. A study can be conducted to analyze its effect on the performance ratio and the corresponding running times. Another interesting future work could be to perform traffic engineering under protection where nodes and link failures are considered. Introduction of new redundant constraints in the formulation could also affect the performance metrics and could be taken up as a research question.

Bibliography

[1] T. D. Nadeau, *MPLS network management: MIBs, tools, and techniques.* Elsevier, 2003.

[2] X. Li and K. L. Yeung, "Traffic engineering in segment routing using MILP," pp. 1–6, IEEE, ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019.

[3] H. Shin, J. Jung, and Y. Koo, "Forecasting the video data traffic of 5G services in South Korea," *Technological Forecasting and Social Change*, vol. 153, p. 119948, 2020.

[4] H. Tahaei, F. Afifi, A. Asemi, F. Zaki, and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *Journal of Network and Computer Applications*, vol. 154, p. 102538, 2020.

[5] D. Jiang and G. Liu, "An overview of 5G requirements," *5G Mobile Communications*, pp. 3–26, 2017.

[6] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," 2002.

[7] J. H. Salim and U. Ahmed, "RFC2884: Performance evaluation of explicit congestion notification (ECN) in IP networks," 2000.

[8] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Power Engineering Review*, vol. 9, no. 4, pp. 101–102, 1989.

[9] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network*, vol. 14, no. 2, pp. 28–33, 2000.

[10] M. R. Abbasi, A. Guleria, and M. S. Devi, "Traffic engineering in software defined networks: a survey," *Journal of Telecommunications and Information Technology*, 2016.

[11] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture," 2018.

[12] F. Hao, M. Kodialam, and T. Lakshman, "Optimizing restoration with segment routing," pp. 1–9, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, 2016.

[13] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 779–792, 2016.

[14] E. Moreno, A. Beghelli, and F. Cugini, "Traffic engineering in segment routing networks," *Computer Networks*, vol. 114, pp. 23–31, 2017.

[15] Y. Zhao, S. Wang, S. Xu, X. Wang, X. Gao, and C. Qiao, "Load balance vs energy efficiency in traffic engineering: A game theoretical perspective," pp. 530–534, IEEE, 2013 Proceedings IEEE INFOCOM, 2013.

[16] K. Long, Z. Zhang, and S. Cheng, "Load balancing algorithms in MPLS traffic engineering," pp. 175–179, 2001 IEEE Workshop on High Performance Switching and Routing., 2001.

[17] T. Ye, S. Kalyanaraman, K. Vastola, *et al.*, "Minimizing packet loss by optimizing ospf weights using online simulation," pp. 79–86, 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems., 2003.

[18] W. Sun, P. Bhaniramka, and R. Jain, "Quality of service using traffic engineering over MPLS: An analysis," pp. 238–241, Proceedings 25th Annual IEEE Conference on Local Computer Networks. LCN 2000, 2000.

[19] E. Amaldi, A. Capone, and L. G. Gianoli, “Energy-aware IP traffic engineering with shortest path routing,” *Computer Networks*, vol. 57, no. 6, pp. 1503–1517, 2013.

[20] Y. Wang and Z. Wang, “Explicit routing algorithms for internet traffic engineering,” pp. 582–588, IEEE, Proceedings Eight International Conference on Computer Communications and Networks), 1999.

[21] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, “A survey on the contributions of software-defined networking to traffic engineering,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 918–953, 2016.

[22] R. Malekian and A. H. Abdullah, “Traffic engineering based on effective envelope algorithm on novel resource reservation method over mobile internet protocol version 6,” *arXiv preprint arXiv:1211.5221*, 2012.

[23] Z. N. Abdullah, I. Ahmad, and I. Hussain, “Segment routing in software defined networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 464–486, 2018.

[24] A. Bashandy, C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, “Segment Routing with MPLS data plane,” *draft-ietf-spring-segmentrouting-mpls-14 (work in progress), IETF*, 2018.

[25] C. Filsfils, D. Dukes, S. Previdi, J. Leddy, S. Matsushima, and D. Voyer, “IPv6 segment routing header (SRH),” *IETF InternetDraft, available: https://tools. ietf. org/html/draft-ietf-6man-segment-routing-header-18*, 2019.

[26] S. Previdi, C. Filsfils, A. Lindem, A. Sreekantiah, and H. Gredler, “Segment routing prefix segment identifier extensions for bgp,” IETF RFC 8669, 2019.

[27] R. Guedrez, O. Dugeon, S. Lahoud, and G. Texier, “Label encoding algorithm for mpls segment routing,” pp. 113–117, 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), 2016.

[28] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, and P. Castoldi, “Efficient label encoding in segment-routing enabled optical networks,” pp. 34–38, IEEE,

2015 International Conference on Optical Network Design and Modeling (ONDM), 2015.

[29] P. L. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, and F. Clad, “Segment routing: a comprehensive survey of research activities, standardization efforts, and implementation results,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 182–221, 2020.

[30] S. Even, A. Itai, and A. Shamir, “On the complexity of time table and multi-commodity flow problems,” pp. 184–193, IEEE, 16th Annual Symposium on Foundations of Computer Science, 1975.

[31] D. B. Johnson, “A note on dijkstra’s shortest path algorithm,” *Journal of the ACM (JACM)*, vol. 20, no. 3, pp. 385–388, 1973.

[32] B. Wu and K. L. Yeung, “Monitoring cycle design for fast link failure detection in all-optical networks,” pp. 2315–2319, IEEE, IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference, 2007.

[33] V. M. Vokkarane and J. P. Jue, “Prioritized routing and burst segmentation for qos in optical burst-switched networks,” pp. 221–222, IEEE, Optical Fiber Communication Conference and Exhibit, 2002.

[34] F. Durand and T. Abrao, “Energy efficiency analysis in adaptive fec-based lightpath elastic optical networks,” *Journal of Circuits, Systems and Computers*, vol. 24, no. 09, p. 1550133, 2015.

[35] A. Giorgetti, N. Sambo, I. Cerutti, N. Andriolli, and P. Castoldi, “Label preference schemes for lightpath provisioning and restoration in distributed gmpls networks,” *Journal of Lightwave Technology*, vol. 27, no. 6, pp. 688–697, 2009.

[36] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2021.