Approximation algorithm design and analysis in next generation communication networks

Haitao Wu

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : Informatique

École Doctorale 536 « Sciences et Agronomie »
Laboratoire d'Informatique (EA 4128)

Conception et analyse d'algorithmes d'approximation dans les réseaux de communication de nouvelle génération

par
Haitao Wu

Soutenue publiquement 05 Novembre 2018 devant un jury composé de :

M.	Walid Ben-Ameur	Professeur, Télécom SudPairs	Rapporteur
M.	Ken Chen	Professeur, Université de Pairs 13	Rapporteur
M.	Eitan Altman	Directeur de recherche, INRIA Sophia Antipolis	Examinateur
M.	Abderrahim Benslimane	Professeur, Université d'Avignon	Examinateur
Ms.	Annie Gravey	Professeur, IMT Atlantique	Examinateur
M.	Juan-Manuel Torres-Moreno	Maître de Conférences (HDR), Université d'Avignon	Directeur
M.	Yaojun Chen	Professeur, Université de Nanjing	Co-directeur
M.	Fen Zhou	Maître de Conférences (HDR), Université d'Avignon	Co-directeur

Laboratoire d'Informatique d'Avignon

ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THESIS

A thesis submitted at the University of Avignon
for the degree of Doctor of Philosophy

In : Computer Science

Doctoral School 536 «Sciences et Agroscience»
Laboratoire d'Informatique (EA 4128)

Approximation algorithm design and analysis in next generation communication networks

by

Haitao Wu

Defended publicly November 5th 2018 before the jury members:

M.	Walid Ben-Ameur	Professor, Télécom SudPairs	Reviewer
M.	Ken Chen	Professor, University of Pairs 13	Reviewer
M.	Eitan Altman	Senior researcher, INRIA Sophia Antipolis	Examiner
M.	Abderrahim Benslimane	Professor, University of Avignon	Examiner
Ms.	Annie Gravey	Professor, IMT Atlantique	Examiner
M.	Juan-Manuel Torres-Moreno	Associate Professor (HDR), University of Avignon	Supervisor
M.	Yaojun Chen	Professor, Nanjing University	Co-supervisor
M.	Fen Zhou	Associate Professor (HDR), University of Avignon	Co-supervisor

Laboratoire d'Informatique d'Avignon

Résumé

Avec l'avènement de l'ère intellectuelle et de l'Internet of Everything (IoE), les besoins de la communication mondiale et des applications diverses ont explosé. Cette révolution exige que les futurs réseaux de communication soient plus efficaces, intellectuels, agiles et évolutifs. De nombreuses technologies réseau sont apparues pour répondre à la tendance des réseaux de communication de nouvelle génération tels que les réseaux optiques élastiques (EONs) et la virtualisation de réseau. De nombreux défis apparaissent avec les apparences de la nouvelle architecture et de la nouvelle technologie, telles que le routage et l'allocation de ressource spectrale (RSA) dans les EONs et l'intégration de réseaux virtuels (Virtual Network Embedding ou VNE) dans la virtualisation de réseau.

Cette thèse traite la conception et l'analyse d'algorithmes d'approximation dans trois problèmes d'optimation du RSA et du VNE : les impacts de la distribution du trafic et de la topologie du réseau sur le routage tout optique, de l'allocation de ressource spectrale, et du VNE dans les topologies des chemins et cycles.

Pour le routage tout optique, le premier sous-problème du RSA, il y a toujours un problème en suspens concernant l'impact de la distribution du trafic et de la topologie EON. Comme le routage tout optique joue un rôle essentiel pour la performance globale de la RSA, cette thèse fournit une analyse approfondi théorique sur ces impacts.

Pour le deuxième sous-problème du RSA, l'allocation de ressource spectrale, deux chemins optiques quelconques partageant des fibres optiques communes pourraient devoir être isolés dans le domaine spectral avec une bande de garde appropriée pour empêcher la diaphonie et / ou réduire les menaces de sécurité de la couche physique. Cette thèse considère le scénario dans lequel les exigences de bandes de garde réelles optiques sont différentes pour différentes paires de chemins, et étudie comment affecter les ressources spectrales efficacement dans une telle situation.

L'hétérogénéité de la topologie des demandes de réseau virtuel (VNR) est un facteur important qui entrave les performances de la VNE. Cependant, dans de nombreuses applications spécialisées, les VNR ont des caractéristiques structurelles communes *par exemple*, des chemins et des cycles. Pour obtenir de meilleurs résultats, il est donc essentiel de concevoir des algorithmes dédiés pour ces applications en tenant compte des caractéristiques topologiques. Dans cette thèse, nous prouvons que les problèmes VNE dans les topologies de chemin et de cycle sont NP-difficiles. Afin

de les résoudre, nous proposons des algorithmes efficaces également analysons leurs ratios d'approximation.

Mots-clés: Réseaux optiques élastiques (EONs), Routage et allocation de ressource spectrale (RSA), Allocation de ressource spectrale sous contrainte de distance (DSA), Virtual Network Embedding (VNE), Algorithme d'approximation

Abstract

With the coming of intellectual era and Internet of Everything (IoE), the needs of worldwide communication and diverse applications have been explosively growing. This information revolution requires the future communication networks to be more efficient, intellectual, agile and scalable. Many technologies have emerged to meet the requirements of next generation communication networks such as Elastic Optical Networks (EONs) and networking virtualization. However, there are many challenges coming along with them, such as Routing and Spectrum Assignment (RSA) in EONs and Virtual Network Embedding (VNE) in network virtualization. This dissertation addresses the algorithm design and analysis for these challenging problems: the impacts of traffic distribution and network topology on lightpath routing, the distance spectrum assignment and the VNE problem for paths and cycles.

For lightpath routing, the first subproblem of the RSA, there is always a pending issue that how the changes of the traffic distribution and EON topology affect it. As the lightpath routing plays a critical role in the overall performance of the RSA, this dissertation provides a thoroughly theoretical analysis on the impacts of the aforementioned two key factors. To this end, we propose two theoretical chains, and derive the optimal routing scheme taking into account two key factors.

We then treat the second subproblem of RSA, namely spectrum assignment. Any two lightpaths sharing common fiber links might have to be isolated in the spectrum domain with a proper guard-band to prevent crosstalk and/or reduce physical-layer security threats. We consider the scenario with diverse guard-band sizes, and investigate how to assign the spectrum resources efficiently in such a situation. We provide the upper and lower bounds for the optimal solution of the DSA, and further devise an efficient algorithm which can guarantee approximation ratios in some graph classes.

The topology heterogeneity of Virtual Network Requests (VNRs) is one important factor hampering the performance of the VNE. However, in many specialized applications, the VNRs are of some common structural features *e.g.*, paths and cycles. To achieve better outcomes, it is thus critical to design dedicated algorithms for these applications by accounting for topology characteristics. We prove the $\mathcal{NP}$-Harness of path and cycle embeddings. To solve them, we propose some efficient algorithms and analyze their approximation ratios.

Key-words: Elastic Optical Networks (EONs), Routing and Spectrum Assignment

(RSA), Distance Spectrum Assignment (DSA), Virtual Network Embedding (VNE), Approximation Algorithms.

Acknowledgements

This work is supported by China Scholarship Council and French Eiffel Scholarship.

First and foremost, I offer my gratitude to Dr. Fen Zhou, who supported me throughout my thesis with his guidance and encouragement. Without his effort, patience and availability during these three years, I would not have been able to complete this work.

I would like to thank Prof. Yaojun Chen and Prof. Juan-Manuel Torres-Moreno. They always give me very good advices and encourage me in my work. I am very grateful to them. I appreciate Prof. Yaojun Chen to encourage me to participate in the joint-supervision program.

I would like to thank Prof. Zuqing Zhu for his invaluable suggestions and kind encouragement. I appreciate the opportunity to work with him.

I also would like to thank my committee members for their prompt evaluation and comments to improve the quality of this dissertation.

I would like to thank my friends Min Ju, Jingyi Jiang, Weihua Wu and all the fellow graduate students in University of Avignon for their friendship during the three years of my stay in France. Special thanks to Min Ju and Weihua Wu for the opportunity of working together and valuable discussion.

I would like to thank fellow graduate students and friends Yanbo Zhang, Xiaolan Hu, FangFang Zhang, Xuemei Zhang, Longqin Wang in Nanjing University for their friendship during my work.

Last but not least, I would like to thank my family and xiaowanzi for their encouragement and support throughout my life.

Introduction

Nowadays, the trend of Internet, especially driven by Big data applications and Artificial Intelligence (AI) [46, 47], marches towards involving more network elements and end-users, larger volume of traffic, and more various applications. The demands of traffic bandwidths are growing exponentially and the types of communication requests are exploding aggressively. To meet the needs of the next generation communication networks, many advanced network architectures and technologies such as Elastic Optical Networks (EONs) and network virtualization arise rapidly.

On the one hand, to address the rapid growth of traffic demands in backbone networks, how to utilize the spectral resources in optical fibers efficiently and intelligently has become a key challenge for all-optical networks. To address this challenge, flexible-grid EONs have been proposed to enhance the agility of bandwidth allocation in the optical layer [35, 83]. Specifically, by the bandwidth-variable transponders (BV-Ts) and wavelength-selective switches (BV-WSS'), EONs can establish lightpaths with several narrow-band (*i.e.*, 12.5 GHz) and spectrally-contiguous frequency slots (FS) and realize data transmissions over them [46]. Therefore, with the fine bandwidth allocation granularity of an FS, EONs can offer just-enough bandwidth to traffic demands from upper-layer networks [37, 71]. The spectrum utilization thus can be effectively improved in EONs in contrast to the traditional fixed-grid Wavelength Division Multiplexing (WDM) optical networks whose channel sizes are usually at 50 GHz-100 GHz.

On the other hand, to prop up multi-types network elements, Internet infrastructure should support new networking mechanisms and applications timely. However, current Internet infrastructures, consisting of a variety of technologies to run distributed protocols, become a barrier to the improvement of Internet service. This diversification is often referred to as the Internet ossification problem [67]. Network virtualization has been regarded as a compelling tool to overcome the Internet ossification and attracting a lot of researches [10, 17, 25, 28, 33, 77]. In the paradigm of network virtualization, the role of traditional Internet Service Providers (ISPs) is separated into two new entities: Infrastructure Provider (InP) and Service Provider (SP). The InP owns and manages the substrate network while the SP focuses on offering customized services to clients. Thus, network virtualization supports various networks of diverse natures (*e.g.*, network architectures, protocols, and user interactions [77]) to coexist in a same substrate network and share substrate resources (*e.g.*, CPUs and bandwidths).

Coming along with these novel architectures and network technologies are many

pristine and complicated problems for service provisioning. In EONs, the fundamental problem is the Routing and Spectrum Assignment (RSA) [21, 30], *i.e.*, how to route a connection request from its source to its destination by a lightpath, and then assign a block of available FSs on it. The RSA problem has been proven $\mathcal{NP}$-hard in [21], which naturally consists of two subproblems: (i) lightpath routing, and (ii) spectrum assignment. More specially, in the lightpath routing, a request should be routed by setting up an appropriate lightpath to connect its source to its destination. While the spectrum assignment, relatively analogous to the graph coloring problem [75], is to allocate available FSs on the lightpath. In network virtualization, how to effectively allocate resources of the substrate network to Virtual Network Requests (VNRs) is a vital problem, which is often referred to as the Virtual Network Embedding (VNE) problem [25]. Explicitly, the VNE needs to find a Substrate Node (SN) (a substrate path, respectively) to meet the computing requirement of each VN (the bandwidth requirement of each VL, respectively) of a VNR. The former is also called *Node Mapping* and the latter is named *Link Mapping*. This thesis revolves closely around the algorithm design and analysis in the three important topics: the lightpath routing, spectrum assignment and the VNE problem to address some challenges appearing in communication networks.

The rest of this chapter is organized follows:

- Motivation and Objectives
- Contributions of the Thesis
- Organization of the Thesis

Motivation and Objectives

The lightpath routing is critical to the final performance of the RSA, and many routing schemes have been proposed [16, 49, 52, 71]. The performance evaluations for the most of the existing routing schemes have been conducted only through numerical simulations. However, the numerical results are significantly influenced by the assumptions on traffic distributions and EON topologies. On the one hand, given a same EON, the performances of a routing scheme can be different under different traffic distributions. On the other hand, given a same traffic distribution, the performance of a routing scheme can also vary for EON with different topologies. Further, when the EON topology and traffic distribution are given, how to take into account the two factors to optimize the lightpath routing is desired. Thus, it is pivotal to figure out how the traffic distribution and EON topology impact on the lightpath routing by deeply theoretical studies. By leveraging random graph theory, this thesis studies the synthesized impact of the two factors on the lightpath routing of the RSA problem.

The spectrum assignment for a set of requests in EONs needs more delicate works to mitigate the interferences among requests than that in a WDM network. More specially, after routing all requests on lightpaths, for any two requests whose lightpaths share some common fiber link(s), the FSs assigned to them should be separated by a guard band in the spectrum domain. In the literature [37, 44, 60, 82], the guard bands

are set as the same frequency size. However, the interference levels between different request pairs vary accordingly. Therefore the guard band with the constant size can not adapt to the changeable interferences and will cause a waste of spectrum resources since the frequency size of the guard band is usually set big enough so as to mitigate all possible interferences. This thesis studies a new spectrum assignment model, where guard bands with different sizes can adapt to the interference levels in an EON. This model is named as distance spectrum assignment (DSA). The DSA tries to achieve efficient spectrum assignment that can not only use guard bands with various sizes to adapt to the mutual interference levels of the lightpaths, but also minimize the maximum FS index used in EONs.

The VNE, as the main challenging problem in network virtualization, drew a lot of attentions of researchers [10, 25]. To address this extremely hard problem, many solutions have been proposed in the literature [17, 28, 33, 51, 77, 81] including heuristics and Integer Linear Programming (ILP), *etc*. One of the key impediments to solve the VNE problem is the topology heterogeneity of both VNRs and substrate networks [10]. However, this is not always true in many specific applications and substrate networks. For instance, the topologies of network service chains are paths [38], and there are many optical rings (*i.e.*, cycles) [61]. For these applications and infrastructures, specialized cloud service providers outperforming the general SPs are desired, which can afford dedicated algorithms taking into account the topology characteristics of the VNRs and substrate networks. Besides, paths and cycles are two of the most fundamental topologies in network structures. Exploiting the characteristics of path and cycle embeddings is vital to tackle general topology embedding. In this thesis, we systematically, from theoretical hardness analysis to practical algorithm design, investigate the VNE problem for the two special topologies.

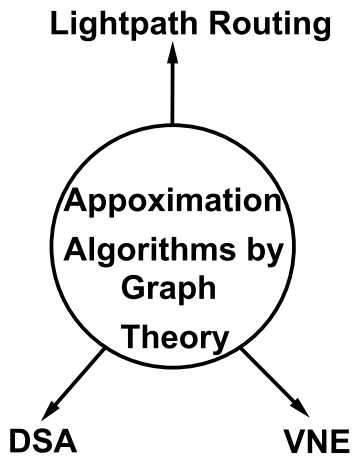


Figure 1: *The skeleton of this thesis.*

Contributions of the Thesis

This thesis investigates the algorithm design and analysis for some challenging problems in next generation communication networks: RSA in EONs and VNE in network virtualization. The studied problems are very challenging and generally $\mathcal{NP}$-hard. The techniques based on graph theory are very powerful and helpful to model and then solve them. To this end, we proposed some approximation algorithms and gave some theoretical analysis on them by leveraging graph theory as shown in Fig. 1. The contributions of this thesis are listed and explained in detail below.

1. The synthesized impact of the traffic distribution and network topology on the lightpath routing in EONs: For this topic, we first provide a theoretical analysis about the impact of the lightpath routing on the overall performance of RSA. We prove that the optimality of the RSA can be measured by the chromatic number of the conflict graph. More specifically, the conflict graph is an auxiliary graph that describes the intersections among the lightpaths for routing requests. Based on the theoretical analysis, we derive several RSA algorithms of constant approximation ratios. We then show the intersecting probability of the lightpath routing determines the chromatic number of its conflict graph by leveraging random graph theory. We further introduce a matrix of conflict coefficients concerning intersection probability, which establishes the connection between the lightpath routing and the traffic distribution as well as EON topology. Finally, we propose a quadratic programming named Global Optimal Formulation (GOF) to derive the optimal routing scheme, which results in the minimum intersecting probability.

2. The distance spectrum assignment in EONs: We prove the $\mathcal{NP}$-hardness of the DSA problem, analyze its inapproximability and also formulate an ILP model to solve it exactly. By connecting with Hamilton path and graph coloring, we formally provide the upper and lower bounds of the optimal solution of the DSA and prove that they are tight. Then, we propose a two-phase algorithm to solve the DSA problem time-efficiently, and study its performance in various situations, which are represented by different conflict graphs. Specifically, in a conflict graph of the DSA, each vertex represents a lightpath while an edge signifies the guard band requirement between two lightpaths. In its first phase, the proposed algorithm generates an initial solution, which is optimal in bipartite conflict graphs and can guarantee an approximate ratio in complete conflict graphs. The second phase improves the initial solution with a random optimization procedure, which proves to be converged.

3. The VNE problem in paths and cycles: For the two special embeddings (path and cycle), we address several important theoretical questions and propose some algorithms. We prove that path embedding problem is still $\mathcal{NP}$-hard even in a simplified model. Especially, the hardness of single path embedding is equivalent to that of determining Supereulerian graph or longest trail problem in two different models. Leveraging Eulerian trail, some approximation algorithms are thus proposed for the first time. We further characterize the topologies of substrate networks which is more suitable for path embedding. For path embedding in realistic scenarios, we give the

inapproximability of path embedding. Then, by connecting with Multiple Knapsack Problem (MKP) and Multi-Dimensional Knapsack Problem (MDKP), efficient and effective MKP-MDKP-based algorithms are developed. For cycle embedding, we propose a Weighted Directed Auxiliary Graph (WDAG) by which we succeed to establish an one-to-one relation between each directed cycle in WDAG and each feasible embedding. Based on that, a polynomial-time algorithm is herein devised to achieve the least-resource-consuming cycle embedding.

Organization of the Thesis

Following the three topics mentioned above, the thesis unfolds corresponding investigations respectively. The rest of the thesis is divided into four parts: Background and Technological Context (**Chapter** 1); On the Impacts of Traffic Distribution and Network Topology on Lightpath Routing in Elastic Optical Networks (**Chapter** 2); On the Distance Spectrum Assignment in Elastic Optical Networks (**Chapters** 3); On Virtual Network Embedding: Paths and Cycles (**Chapter** 4).

In Chapter 1, we give the backgrounds on the EONs and network virtualization respectively. The evolution of the optical networks from the traditional WDM networks to the EONs is presented. Specially, we introduce the RSA problem in EONs and present an important concept in the RSA problem, *i.e.*, the conflict graph. For network virtualization, Chapter 1 profiles the future Internet infrastructure under the business model of Infrastructure as a Service (IaaS) and two important components in network virtualization: the substrate network and VNRs. The VNE problem is also put forward. Besides, the techniques in graph theory have been used through the thesis to solve all the three topics presented. Chapter 1 also provides relevant concepts and useful notations in graph theory such as trail, graph coloring, Hamilton path, *etc*.

Chapter 2 concerns the impact of traffic distribution and network topology on the lightpath routing. We provide a theoretical analysis framework that reveals the relation between the optimality of the lightpath routing and the chromatic number of its conflict graph. Based on this theoretical deduction, some approximation algorithms are presented. By random graph theory, we then introduce the connection between intersecting probability and chromatic number, and propose the conflict coefficients and a quadratic programming, which embody the impact of the traffic distribution and network topology. Within the proposed theoretical analysis, we evaluate three EONs under two traffic distributions. Extensive simulations are conducted to verify our analysis framework.

Chapter 3 addresses the DSA problem. We formally model the DSA problem in an ILP and analyze its $\mathcal{NP}$-hardness and inapproximability. The upper and lower bounds of the optimal solution of the DSA are analyzed by leveraging Hamilton path and graph coloring, which prove to be tight. We then transform the DSA into a Permutation-based Optimization Problem (POP). With this transformation, a two-phase algorithm is developed. The performance of the two-phase algorithm is theoretically analyzed. The first phase outputs an initial solution, which can guarantee approximation ratios in some

special graph classes, and the second phase improves the quality of the initial solution by a random approach whose convergence is also proved. The numerical results show that the proposed algorithm approaches the optimal solution well. The work of this chapter can be found in the publications [J1, C1].

Chapter 4 treats the VNE problems in two special topologies, paths and cycles. We present the network models and the formal description of the VNE problem. As for path embedding, we put forward a preliminary model to help understand its characteristics. The proof of $\mathcal{NP}$-hardness and some approximation algorithms are provided in the preliminary model. For realistic scenarios, we present the inapproximability result and devise the MKP-MDKP-based algorithms. For cycle embedding, we elaborate the construction of WDAG, bear out the one-to-one relation between directed cycles and feasible embedding ways, and also craft the specialized cycle-embedding algorithm. We conduct simulations under different scenarios to demonstrate the superiority of our proposed algorithms over existing ones in the two special topologies.

Finally, **Chapter** 5 concludes this thesis and envisions future work.

Contents

Chapter 1

Background and Technological Context

Contents

1.1 Introduction

In this chapter, we introduce the challenging RSA problem in EONs and the VNE problem in network virtualization. To solve these challenging problems, optimization techniques based on graph theory are required. Thus, some concepts in graph theory used in this thesis are also presented.

The organization of this chapter is as follows.

- In Section 1.2, we introduce the advantage of EONs over the traditional WDM networks and expatiate related works of the RSA problem.
- In Section 1.3, we outline the business model of network virtualization and summarize related work in the VNE problem.
- In Section 1.4, some concepts in graph theory are introduced.

1.2 The Background and RSA problem in EONs

1.2.1 Evolution from fixed-grid WDM to flexible-grid EON in optical networks

In the last decade, due to the proliferation of various types of bandwidth-hungry applications such as IPTV, video-ondemand, inter-datacenter networking, *etc*, Internet traffic is explosively growing [16, 57, 65, 66]. Cisco predicted that Internet traffic will increase nearly threefold over the next 5 years [1]. Optical networking technologies are crucial to the operation of the global Internet and its ability to support critical and reliable communication services [65]. Optical networks have the potential to support the continued bandwidth demands [16]. According to the expectation of TeleGeography that international bandwidth demands will be approximately 606.6 Tb/s in 2018 and 1,103.3 Tb/s in 2020 [16], optical networks will be required to support Tb/s order transmission in the near future [36]. However, traditional WDM networks are incompetent of scaling performance to meet the growing traffic demands. The technology of WDM networks follows the standardized ITU-T fixed frequency grids [2] of 50 GHz or 100 GHz. Under this coarse granularity of channel size, spectrum resources will hugely wast if a complete optical channel is allocated even when the data rate of an upper-layer application can only partially occupy the whole capacity pipe of a WDM optical channel [57] as shown in Fig. 1.1.

Whereas the limitation of fixed-grid of conventional WDM networks, which dose not adapt to bandwidth heterogeneity of future traffics, the concept of flexible-grid EON has been introduced to improve the utilization efficiency of spectrum resources [37]. With the advances in Orthogonal Frequency-Division Multiplexing (OFDM), by BV-Ts and BV-WSS', an EON has the ability to dynamically adjust its channel size in an optimum and elastic way according to the continuous varying traffic bandwidth demands [66]. More specifically, the optical spectrum of a fiber link in EONs is divided

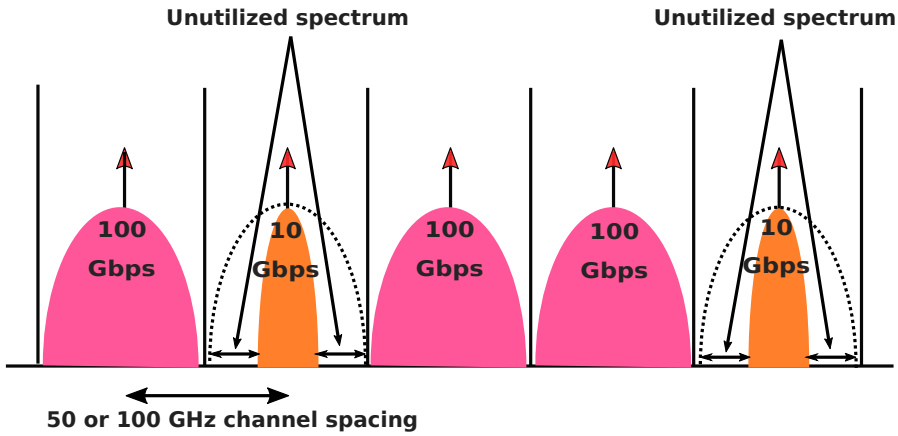


Figure 1.1: *IUT-T fixed-grid [16].*

into narrow FSs, having a spectral width of 12.5 GHz (or less), and the channel can be a combination of a set of continuous FSs as shown in Fig. 1.2. Thus, with the fine-granularity and flexible channel size, the spectrum utilization in EONs is greatly improved compared to WDM networks. With advanced elastic optical components and techniques such as BV-Ts and BV-WSS', EONs will boost the next generation optical networks moving towards to the goals of greater efficiency flexibility, scalability [16]. Therefor, the EON architecture is considered as one of the most promising candidates for next generation optical networks.

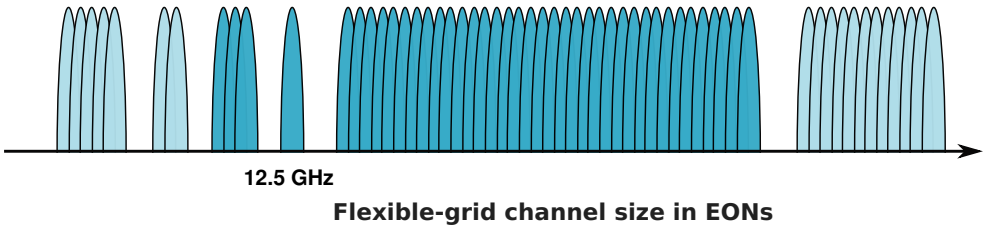


Figure 1.2: *Flexible-grid in EONs [16].*

1.2.2 The RSA problem in EONs

Nevertheless, EONs not only bring in the benefits but increase the complexity of network planning. The RSA problem in EONs is much more challenging than its counterpart Routing and Wavelength Assignment (RWA) in WDM networks for two reasons: (1) the spectrum contiguity; (2) the channel size heterogeneity.

In this thesis, an EON is usually denoted by a connected graph $G(V, E)$, where V represents the set of underlying nodes and E is the set of fiber links as shown in Fig.

1.3. Note that usually there are two fibers in each link and each fiber is in charge of one opposite communication. All fiber links have the same set of spectrum resources.

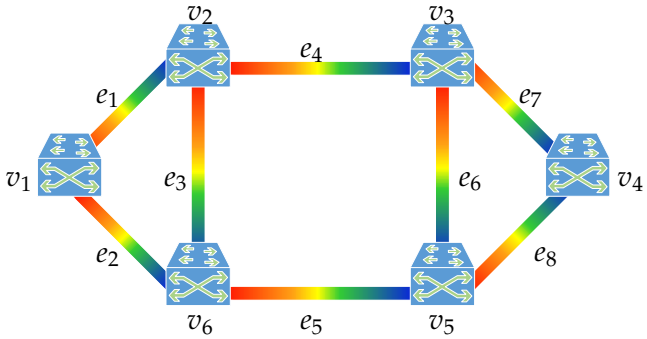


Figure 1.3: *A 6-node-and-8-link EON.*

The spectrum resource in each fiber link are measured by the number of FSs as shown in Fig. 1.4.

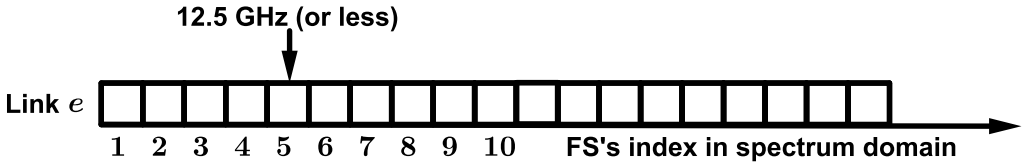


Figure 1.4: *The spectrum resource in a fiber link e.*

A communication request R in the EON $G(V, E)$ is represented by a 3-tuple (s, d, R^w), where $s \in V$ and $d \in V$ are the source and destination node respectively and R^w is the bandwidth requirement in terms of a number of FSs. When a set of requests $\mathcal{R}$ arrive, the EON needs to find an appropriate lightpath to connect the source and destination of each request and allocate suitable FSs on the routed lightpath, which is also known as the RSA problem [75].

The RSA is the fundamental problem in EONs for service provisioning. In the study of the RSA problem, we generally take into account the following constraints.

- **Bandwidth Requirement Constraint**. The number of FSs assigned to each request should be no smaller than its bandwidth requirement.
- **Spectrum Contiguity Constraint**. The FSs assigned to a request must be contiguous in the spectrum domain. This is a physical layer constraint for all-optical communications in EONs.
- **Spectrum Continuity Constraint**. All the fiber links on the lightpath for a request should be assigned with the same set of contiguous FSs.
- **Guard Band Constraint**. To mitigate mutual interference, when two lightpaths share some fiber link(s), the FSs assigned to the two requests should be separated in the spectrum domain by a guard band frequency.

With respect to the objective function of the RSA problem, two variants have been studied in the literature: max-RSA and min-RSA [43, 69]. The former problem focuses on the service provisioning in EONs with limited spectrum resources, and the objective is to maximize the number of requests that can be served. The latter one has a planning concern and its objective is to minimize the required spectrum resource to route all the requests [43, 69]. In this thesis, we focus on the latter one with planning concern.

The RSA problem, which prove to be $\mathcal{NP}$-hard [21], can be naturally separated into two subproblems: (i) lightpath routing and (ii) spectrum assignment.

Lightpath Routing

Many routing schemes have been proposed in the literature [41, 49, 52, 64] to address the lightpath routing. To search suitable lightpaths between source-destination pairs, these approaches can be mainly classified into four types [16]: (i) fixed routing, (ii) fixed alternative routing, (iii) least congested routing, and (iv) adaptive routing.

1) Fixed Routing: For each source-destination pair, the fixed routing scheme [49, 52] precomputes a single fixed lightpath by some shortest path algorithm such as Dijkstra's algorithm. When a request arrives in the EON, this scheme attempts to establish a lightpath along the predetermined fixed route. Then, it checks whether there are enough available FSs on each link of the predetermined lightpath to satisfy the request. If even one link does not have the desired FSs, the request is blocked, *i.e.*, can not be served.

2) Fixed Alternative Routing: The fixed alternative routing [49, 52] can be considered as a revision or improvement of the fixed routing scheme. The fixed alternative routing precomputes several candidate lightpaths for each source-destination pair by some K-shortest paths algorithms such as [76]. When a request arrives, this scheme attempts to establish a lightpath through checking each of the candidate routes in sequence, until a route with the required FSs is found. If there does not exist such a lightpath, the request is blocked. Although the computation complexity of this scheme is higher than that of the fixed routing, its blocking probability is comparatively lower than the fixed routing scheme.

3) Least Congested Routing: Similar to the previous one, the least congested routing [16, 49, 52] first predetermines several candidate lightpaths for each source-destination pair. The difference is that when a request arrives, this scheme estimates the congestion on each candidate lightpath, which is measured by the number of FSs available on the each link of the candidate lightpath and selects the least one. The fewer the available FSs are on a candidate lightpath, the more congested it is. While the computation complexity of the least congested routing is higher than the fixed alternative routing, its blocking probability is almost the same as that of the fixed alternative routing scheme.

4) Adaptive Routing: In the adaptive routing [41, 52] the lightpath between the source-destination pair of the request is selected dynamically, relying on the link-state information of the EON. The link-state information of the EON is determined by the set of all requests that are currently served. The most acceptable form of the adaptive

routing is the adaptive shortest path routing, which sets each unused FS in a fiber link as a cost of 1 unit, whereas the cost of each used FS in the link is taken to be $\alpha > 1$. When a request arrives, the shortest path between a source-destination pair is selected.

From the introductions of the above four types of lightpath routing schemes, the main intention of routing schemes is to select an appropriate lightpath for each request and properly dispense all the requests on the candidate lightpaths so that as many requests can be served as possible, which is the center challenge in lightpath routing. Apparently, the lightpath routing is deeply influenced by the traffic distribution and underlying topology of the EON. **Motivated by this fact, how to take into account the two factors to devise a better routing scheme is the first problem that we tend to address in this thesis.**

Besides the above four types of routing schemes, there also exist many variants of lightpath routing considering other aspects such as elastic characteristics, spectrum fragmentations, *etc* [16].

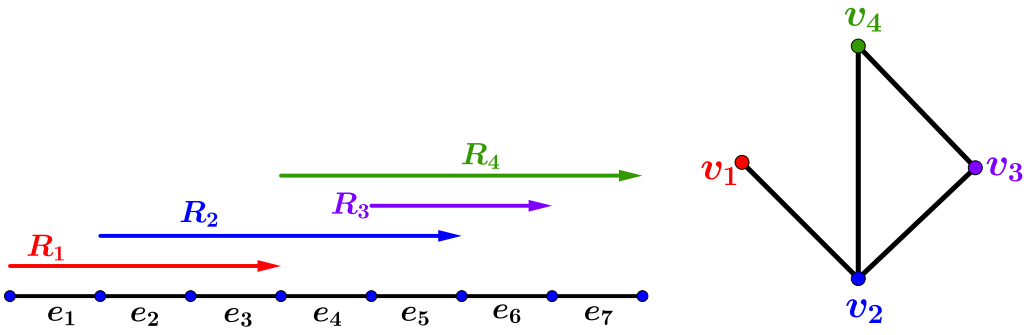
Spectrum Assignment

When a set of requests arrive in an EON, after the lightpath routing phase, all requests are routed on their own lightpaths. Since in this thesis, we consider the RSA problem with planning concern, the spectrum assignment is to assign FSs to each routed lightpath while satisfying the four constraints above. The objective is to minimize the maximum FS index used. Among the approaches to spectrum assignment, leveraging graph-coloring together with conflict graph is an important one [78]. First, we introduce the important concept of conflict graph.

A *conflict graph* [78] $G(V, E)$ is such a graph whose vertex set represents the set of requests in the EONs. For example, each v_i in Fig. 1.5(a) represents the R_i in Fig. 1.5(b) where $1 \leq i \leq 4$. Any two vertices are connected by an edge if and only if the lightpaths, on which the requests are routed, share some fiber link(s) as shown in Fig. 1.5. Besides, each vertex is associated with a vertex weight equal to the required bandwidths of the corresponding request.

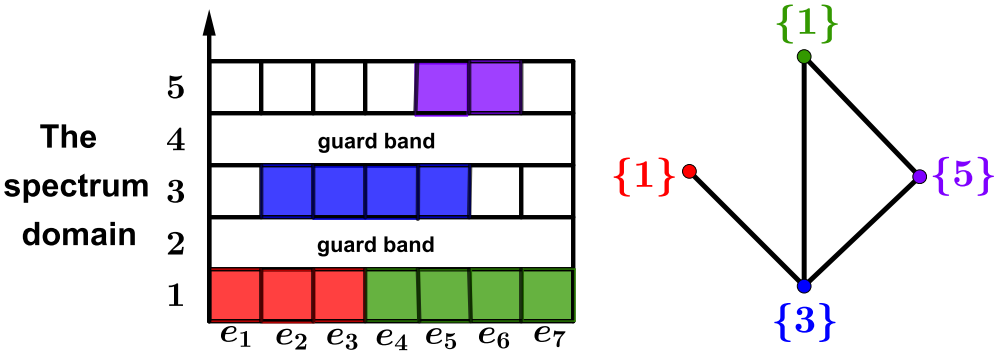
In addition, we can define a *proper spectrum assignment* in the conflict graph as follows: It needs to assign enough FSs to each vertex to meet the vertex weight meanwhile the FSs assigned to two vertices adjacent in the conflict graph must be separated by a guard band frequency appropriately. According to the definition, obviously, any proper spectrum assignment for the conflict graph corresponds to a proper spectrum assignment for the lightpaths, vice versa as shown in Fig. 1.6, where it is assumed that each bandwidth requirement is 1 FS and so as the guard band.

Thus, the conflict graph embodies all the four constraints of RSA mentioned above, and is very important to analyze spectrum assignment. Especially in traditional WDM networks, the chromatic number of the conflict graph is equal to the optimal solution of wavelength assignment [78].



(a) The Routed lightpaths. (b) The Conflict graph.

Figure 1.5: *An example of conflict graph.*



(a) Routed lightpaths. (b) Conflict graph.

Figure 1.6: *The corresponding proper spectrum assignments in the lightpaths of Fig. 1.5(a) and conflict graph of Fig. 1.5(b).*

Revolving around the conflict graph, graph coloring plays an important role in the spectrum assignment. Many graph-coloring-based approaches have been proposed in the literature. In [71], after the construction of the conflict graph, two spectrum assignment algorithms, the Maximum Reuse Spectrum Allocation (MRSA) and Balanced Load Spectrum Allocation (BLSA), are proposed, which are shown to work well through the numerical simulations. [78] reveals many connections between the optimal solution of spectrum assignment and the properties of the conflict graphs. Besides, there exist many other heuristics in the literature [78] such as random, first-fit, least-used, most-used, min-product, *etc*. Therefore, analyzing and leveraging the properties of the conflict graph is a chief way to address the spectrum assignment problem. **To overcome the drawback of constant-size guard band, this thesis analyzes the conflict graph of the DSA problem in which we consider heterogeneous guard band sizes among lightpaths to mitigate mutual interferences.**

All of the RSA problems discussed in this thesis are off-line or static. In addition to the off-line version, there exist many other variants in literature such as on-line RSA

in which we do not have all the informations of the requests at hand, defragmentation, modulation, Quality-of-Transmission (QoT), grooming, survivability, energy saving, filter-less [7, 8, 65], *etc*. A comprehensive survey on the RSA problem can be found in [16].

1.3 Network Virtualization and the VNE problem

1.3.1 The background of network virtualization

Over the past three decades, Internet achieves a stunning success and becomes the core architecture to provide services for global commerce, media, and defense [5, 20, 53]. This utterly changes the way we work, play and learn [5]. Its ability to support diverse distributed applications and a wide variety of network technologies make the ubiquity and centrality of Internet keep increasing [5]. Nonetheless, this popularity becomes the radical barrier to its further growth. Since current Internet deeply depends on a variety of infrastructure technologies to run protocols and distributed applications, this multi-provider nature makes any adopting a new architecture or modification require consensus among competing stakeholders [10, 20]. Developers and researchers have been craving for new architectures that could address many challenges in next generation Internet such as Cloud computing [23]. However, the requirement of Internet Service Providers (ISPs)' joint agreement on any architectural change makes the prospects for significant change in its existing architecture daunting [5]. This resistance of the current Internet to fundamental changes is known as the ossification problem [10]. Even worse, the inability to adapt to new pressures and requirements continues to deteriorate.

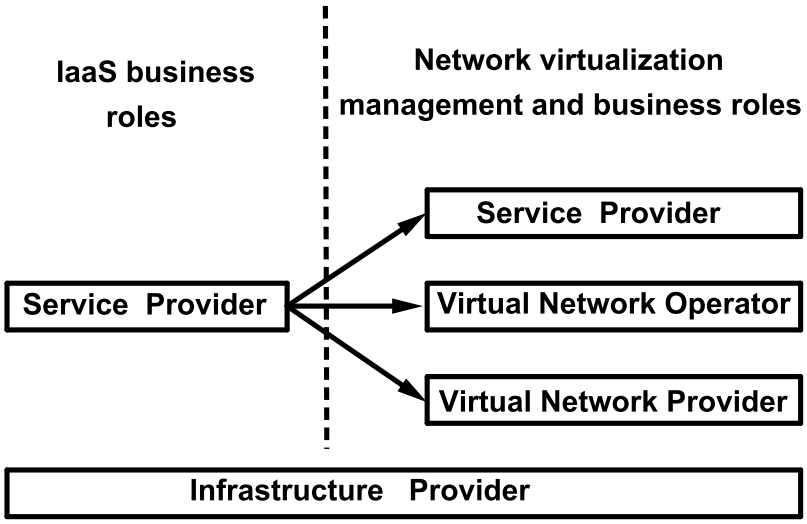


Figure 1.7: *The IaaS business model in network virtualization [25].*

To address the Internet ossification, network virtualization is proposed by the research community and considered as a compelling solution [5, 10, 25]. In network vir-

tualization, future Internet architectures will be based on the Infrastructure as a Service (IaaS) business model [25]. In this model, the role of current ISPs is decoupled into two new entities: Service Provider (SP) and Infrastructure Provider (InP) as shown in the left side of Fig. 1.7. The InP sets up and maintains the physical equipments and substrate resources while the SP is in charge of deploying network protocols and offers end-to-end services [25]. In some research projects such as 4WARD [55], the management and business roles of the SP are further separated into three parts as shown in the right side of Fig. 1.7: The Virtual Network Provider (VNP) which deals with a number of InPs and assembles virtual resources from one or more InPs; the Virtual Network Operator (VNO) which takes care of managing existing virtual networks, and the SP which is free of management and focuses on offering customized services according to demands of clients [10, 25].

In a network virtualization environment, there are two key components: A substrate network and a set of Virtual Network Requests (VNRs) to be served on it.

- **Substrate Network:** A substrate network is owned and operated by an InP which earns a profit from leasing substrate resources to its customers (SPs or VNPs). The physical layer of the substrate network could be all sorts of data information transport networks, both wired *e.g.*, copper, WDM and the EONs mentioned above and wireless. A substrate network consists of a set of Substrate Nodes (SNs) and Substrate Links (SLs). The InP manage kinds of resources and attributes on the substrate network. The computing processes of the VNRs are conducted on the SNs, and thus the CPU capabilities are the most important resources on the SNs. Besides, there usually exit many other attributes on the SNs, *e.g.* locations, storages and protocols *etc*, to satisfy some VNRs of special settings. The intercommunications between different SNs are conducted in the SLs, and thus the bandwidth resources are crucial in each SL. For example, if the substrate network is an EON, then the bandwidth resources are the spectrum resource FSs.
- **Virtual Network Request:** A VNR is a combination of Virtual Nodes (VNs) and Virtual Links (VLs), which is a upper-layer application and constructed by the SP according to the service demands of clients. Each VN usually represents some computing tasks and the VLs deal with the intercommunications between VNs.

After the VNRs are configured pursuant to service demands of clients, the SP needs to discovers resources available in substrate networks by the attributes of InPs and selects appropriate ones for the deployment of VNRs, which is usually referred to as the VNE problem [77].

1.3.2 The VNE problem in network virtualization

The VNE is to deal with the resource allocations both in nodes and links, and thus is the fundamental problem to the service provision in network virtualization. For each VNR served, the VNE needs to (i) select an SN to meet the computing requirement of each VN, and (ii) search a substrate path to satisfy the bandwidth requirement of each VL in a VNR. The former is also called *Node Mapping* and the latter is named *Link Mapping*.

Assuming the SP has constructed a VNR and selected a substrate network to serve the VNR, the constraints of the node mapping and the link mapping of the VNE problem are as follows.

- **Node Mapping Constraint.** For each VN, the VNE needs to select an appropriate SN of enough CPU capability to serve it. Besides, some VNRs have special settings, such as the types of protocols to run and special requirements on the attributes of SNs to serve. For example, in [28], to reduce the latency so as to guarantee the Quality of Experience (QoE), the VNs of VNRs have specific location constraint that they can only be served on specially located SNs. Meanwhile, for each SN, the CPU consumption of the VNs embedded on it can not exceed its capability.
- **Link Mapping Constraint**. Each VL should be embedded into a substrate path which connect the SNs that serve the VNs connected by the VL. Each SL on the embedded substrate path should have enough bandwidth resources to serve the VL.

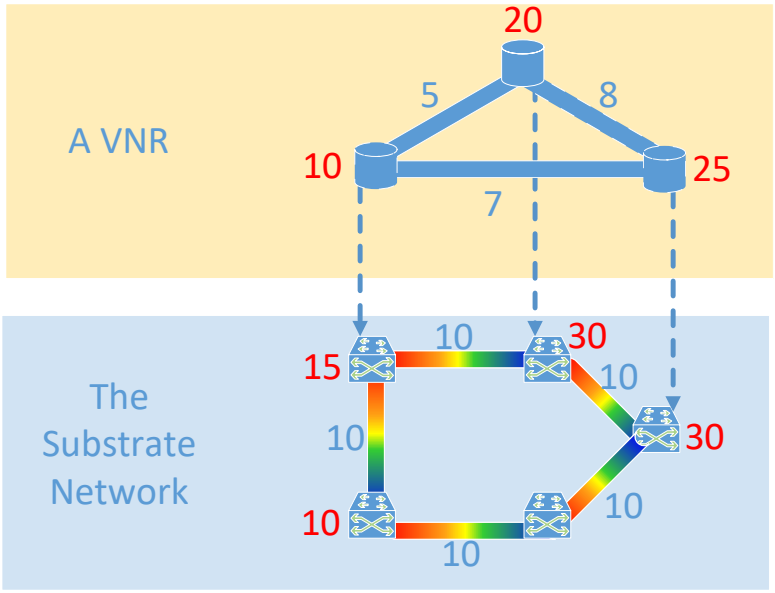


Figure 1.8: *An example of the VNE process.*

Figure 1.8 illustrates an example of the VNE process, where the red numbers indicate the CPU capabilities/requirements of the SNs/VNs respectively and blue numbers beside the SL/VL indicate the bandwidth capabilities/requirements respectively.

In the study of the VNE, there exist many variants for the node mapping and link mapping. For example, for the node mapping, many literatures such as [17, 73, 81], just take into account the CPU requirements of VNs to advance the corresponding researches. In this thesis, we call the generic VNE that only considers the CPU requirement as G-VNE. While in some literatures such as [28], the VNs have other special attribute requirements on the SNs, and we call them S-VNE. The relation between the

G-VNE and the S-VNE will be discussed in Chapter 4. Besides, sometimes embedding many VNs on a same SN makes a VNR more vulnerable to the substrate network failure, some literatures add an additional constraint that two VNs of a same VNR can not be embedded on a same SN. For the link mapping, in [77], the substrate network has the ability to allow substrate path splitting and migrating. Thus, under this link mapping scheme, a VL can be embedded on one or more substrate paths.

For the VNE problem, two main objective functions are generally considered.

- **The Acceptance Ratio:** Given a substrate network and a set of VNRs, the objective is to maximize the number of VNRs that can be embedded on the substrate network. We denote by **AcR** the acceptance ratio problem in this thesis.
- **The Revenue:** Each VNR is associated with a revenue. Its objective is to maximize the total revenue of VNRs that can be embedded on the substrate network. We denote by **Rev** the revenue problem. One may notice that AcR actually is a special case of Rev by setting each revenue as one.

The VNE problem has been proven $\mathcal{NP}$-hard [4]. To address this extremely hard problem, all VNE approaches in the literature can be categorized depending on whether they are static or dynamic, centralized or distributed, and concise or redundant [25] as follows.

- **Static vs Dynamic [25]:** When a set of VNRs arrive, the static approach embed them on the substrate network following a fixed mapping scheme without considering remapping one of more VNRs to improve the performance of the embedding. While in reality, as new VNRs arrive and others expire and, both the substrate resources in SNs and SLs become fragmented. Consequently over the time, the ratio of accepted VNRs diminishes, reducing a long-term revenue. Besides, sometimes, there are some changes and updates taking place in both the topologies of substrate networks and VNRs. The dynamic VNE approaches try to reconfigure the embedded VNRs so as to reorganize the resource allocation and optimize the utilization of substrate resources.
- **Centralized vs Distributed [25]:** There exist two fundamentally different ways to tackle the VNE problem, centralized and distributed. Each method has its own advantages and disadvantages. In a centralized approach, there is one dedicated entity taking the responsibility of embedding. The advantage of this approach is that the entity has the global knowledge about each step of the situation of overall network, and thus facilitates more optimal embeddings. But the centralized is two-edged. If the centralized entity fails, the entire VNE mapping process fails. Besides, the scalability is another problem in large networks, where a single mapping entity may be overwhelmed by the number of VNRs to handle. In contrast, the distributed approach employs multiple entities for the VNE process. By some internal organization, the embedding is distributed among the participating entities. The advantages of such an approach lie at its better scalability and survivability. However, one has to pay for this with synchronization costs. Besides, the lack of global information in each individual entity abut current status

of overall networks may impede the performance of the VNE process.

- **Concise vs Redundant [25]:** The failure of any SNs/SLs will affect all VNs/VLs embedded upon them. Therefore, for some fault-sensitive VNRs, some backup substrate resources should be set up to cope with the failures of SNs and SLs. The concise approaches just utilize as many substrate resources as necessary to satisfy the VNRs. There is no reservation of redundant resources. While a redundant approach reserves additional resources for VNRs in case some substrate resources fail at run-time. In general, the higher the amount of reserved substrate resources and the degree of reliability, the less VNRs can be embedded. A better trade-off should be found between the quality of the reliability and the revenue of embedding.

For the perspective of the methodology, studying the topological characteristics of both the substrate networks and VNRs is an important approach to address the VNE problem. Many topology-aware methods are proposed in the literature. In [17], based on their resource and topological attributes the authors adopt Markov random walk model to rank SNs. According to the ranking, VNs are first embedded onto SNs. Then VLs are embedded between the mapped nodes by finding the shortest paths. [77] rethinks the design of the substrate network. The authors advocate a different approach to allow the substrate network to split VLs and migrate multiple substrate paths. Meanwhile, for some common topology classes, to meet the needs of the key applications, [77] investigates customized embedding algorithms by taking into account the topological characteristics and proves them more effective than general algorithms. In [33], the authors consider embedding VNRs across multiple substrate networks. After splitting VNRs across substrate networks, the VNE problem is solved by using both max-flow min-cut algorithms and linear programming techniques. **Since path and cycle are two fundamental network structures, this thesis investigates the topological characteristics of the corresponding VNE problems and propose effective algorithms to solve them.**

Furthermore, many metrics are used to evaluate the quality of a successful embedding: The Quality of Service (QoS) metrics to measure the impact of an embedding with respect to the service quality when using the VNR, which include many aspects such as path length, stress level, utilization, delay, jitter, *etc*; resource spending metrics; resilience-related metrics and so on. For more details about the VNE problem, one can refer to two comprehensive surveys [10, 25]. In [25], the authors expanded the roles of the SP and InP in the paradigm of network virtualization and proposed a novel classification scheme for current VNE algorithms. The other survey [10] elaborated and emphasized the importance of resource discovery.

1.4 Relevant Concepts in Graph Theory

Graph theory is an important branch of modern applied mathematics. The subject that graph theory studies is a graph $G(V, E)$ consisting of a set V of vertices and a set E of

edges. Definitely, all kinds of communication networks existing in reality, such as the EONs, substrate networks and virtual networks mentioned above, can be characterized by a graph. Thus, this versatility makes graph theory an indispensable tool in the design and analysis of communication networks [13]. The paper "On the Seven Bridges of Konigsberg" written by Euler in 1736 is considered as the first paper in the history of graph theory, in which an important concept , Eulerian trail, was proposed. During the development of graph theory, the Four-Colour Conjecture and Traveling Salesman Problem (TSP) are two core topics [13], which involve many aspects such as graph coloring and Hamilton path. As we shall see, all of the Eulerian trail, graph coloring and Hamilton path play import roles in analyzing and solving the RSA and VNE problems. Here, we introduce relevant graph concepts involved in this thesis in a nutshell as follows.

- **A graph $G(V, E)$:** A *graph* G is an ordered pair (V, E) consisting of a set V of vertices and a set E, disjoint from V, of edges, together with an incidence function f that associates with each edge of E an unordered pair of (not necessarily distinct) vertices of V. If e is an edge and u and v are vertices such that $f(e) = \{u, v\}$, then e is said to connect u and v, and the two vertices u and v are adjacent in G, called the ends of e [13].
- **Degree:** In a graph $G(V, E)$, the *degree* of a vertex v is the number of edges connecting it denoted by d_v. The maximum degree of G is $\max_{v \in V}(d_v)$ denoted by $\Delta(G)$. The number of vertices of odd degrees in G is denoted by $o(G)$.
- **Subgraph:** Given a graph $G(V, E)$, a *subgraph* $G'(V', E')$ is a graph where $V' \subseteq V$ and $E' \subseteq E$.
- **Spanning Subgraph:** Given a graph $G(V, E)$, a *spanning subgraph* $G'(V', E')$ is a graph that $V' = V$ and $E' \subseteq E$.
- **Path:** A *path* is a graph whose vertices can be arranged in a linear sequence such that two vertices are adjacent if they are consecutive in the sequence, otherwise nonadjacent [13].
- **Cycle:** A *cycle* is a graph of three or more vertices whose vertices can be arranged in a cyclic sequence in such a way that two vertices are adjacent if they are consecutive in the sequence, otherwise nonadjacent [13].
- **Hamilton Path:** In a graph G, a *Hamilton path* is a subgraph which is a path traversing all vertices of G. Note that not every graph has a Hamilton path.
- **Hamilton Cycle:** In a graph G, a *Hamilton cycle* is a subgraph which is a cycle traversing all vertices of G. Similarly, not every graph has a Hamilton cycle.
- **Trail:** In a graph $G(V, E)$, a *trail* is such a subgraph that can be expressed as a sequence of vertices "$v_1, v_2, ..., v_n$", where for any two adjacent vertices v_i and v_{i+1}, $1 \leq i \leq n-1$, $v_i v_{i+1}$ is an edge, *i.e.*, $v_i v_{i+1} \in E$ and no repeated edge occurs in the trail, *i.e.*, the pair (v_i, v_{i+1}) only occurs once in the trail. A *closed trail* is such a trail "$v_1, v_2, ..., v_n$", with $v_1 = v_n$. Figure 1.9 illustrates a trail T_1 (left in blue), a

closed trail T_2 (right in red), and the corresponding vertex sequences. The length of a trail is the number of edges on it. For example, in Fig. 7, T_1's length is 6 while T_2's length is 7.

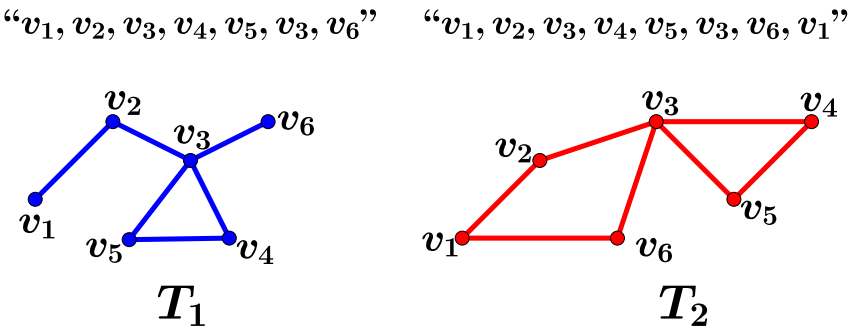


Figure 1.9: *An example of trail and closed trail.*

- **Eulerian trail:** In a graph G, an *Eulerian trail* is a trail traversing all edges. Notice that not every graph has an Eulerian trail. An *Eulerian circuit* is a closed Eulerian trail. Therefore, in Fig. 1.9, if we treat the two trails as two graphs (not subgraphs), then they can also be regarded as an Eulerian trail and an Eulerian circuit respectively.
- **Eulerian graph:** A graph G is called *Eulerian graph* iff it has an Eulerian circuit. In Fig. 1.9, if we treat T_2 as a graph, then T_2 is an Eulerian graph.
- **Supereulerian graph:** A graph G is called *Supereulerian graph* iff it contains a spanning subgraph which is an Eulerian graph. Similarly, if a graph G is formed by adding one edge v_2v_6 to T_2 in Fig. 1.9, *i.e.*, $G = T_2 + v_2v_6$, then G is a Supereulerian graph.
- **Independent Set:** An *independent set* is a set of vertices in a graph, no two of which are adjacent.
- **Clique:** A *clique* is a set of vertices in a graph, every two vertices of which are adjacent.
- **Chromatic Number $\chi(G)$:** Given a graph $G(V, E)$, a *proper graph coloring* is that we need to assign each vertex a color such that any adjacent vertices receive different colors. The minimum number of colors we can use to get a proper graph coloring is the *chromatic number* of G denoted by $\chi(G)$. Since in a proper graph coloring, those vertices receiving a same color must belong to a same independent set, the chromatic number $\chi(G)$ is equal to the minimum number of independent sets into which V can be partitioned.
- **Bipartite Graph:** A *bipartite graph* is a graph whose chromatic number is 2 or whose vertices can be partitioned into two independent sets.
- **Complete Graph:** A *complete graph* is a graph that itself is a clique, *i.e.*, any two vertices in it are adjacent.

- **Perfect Graph:** A *perfect graph* is a graph in which the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph. A perfect graph has many good properties. For instance, its chromatic numbers can be obtained in polynomial time.

1.5 Conclusion

In this chapter, we first introduced related backgrounds of the EONs and network virtualization. Then some necessary concepts in graph theory are presented, which will be helpful to solve the studied networking problems. We presented the RSA problems and the VNE problems, including their constraints and objective functions, and summarized the existing methods to address them. After having a sketched impression on the problems studied in this thesis, we will go in-depth in the next chapters.

Chapter 2

Impacts of Traffic Distribution and Network Topology on Lightpath Routing in Elastic Optical Networks

Contents

2.1 Introduction

As introduced in Chapter 1, the fundamental problem to realize service provisioning in EONs is the RSA problem, *i.e.*, how to route a connection request from its source to its destination by a lightpath, and then assign a block of available FSs on it. Apparently, the lightpath routing is a determinant for the final performance of the RSA. From the introductions of constraints of the RSA problem in Chapter 1, the intersections between the routed lightpaths are most important to the final performance of the RSA problem, since the less intersections, the better optimization in spectrum assignment.

Many routing schemes [16, 49, 52, 71] have been proposed to address the lightpath routing such as: the fixed routing schemes (*e.g.*, shortest paths or other K-shortest paths), and the alternate routing schemes, which, based on some "optimal" principles, *e.g.*, "the least congested" and "the smallest load" *etc*, select the "optimal" path for each request in a set of predetermined candidate paths. Nonetheless, most of current routing schemes are unadaptable to the changes in traffic distributions and EON topologies, and their performance evaluations are mainly based on numerical results rather than rigorous theoretical analysis. However, the numerical results can be easily biased by the choice of traffic distributions and EON topologies. Thus, it is pivotal to figure out how the traffic distribution and EON topology impact the lightpath routing by deep theoretical studies. So that it can provide guidance on how to adapt the routing scheme to achieve the best performance for different network configurations.

In this chapter, to analyze the synthesized impact of the two factors on lightpath routing, we derive two important theoretical chains. In the first theoretical chain, by the chromatic number of the conflict graph and rand graph theory, we provide a theoretical analysis about the impact of the lightpath routing on the overall performance of RSA. Especially, the intersecting probability of lightpaths is pivotal to the optimality of the RSA. In second theoretical chain, we propose a novel concept, conflict coefficients by which the traffic distribution and network topology of the EON decide the intersecting probability. Thus, combining the two theoretical chains, we reveal the synthesized impact. The main contributions of this chapter are summarized as follows.

- We give the upper and lower bounds of the optimal solution of the lightpath routing by analyzing the chromatic number of its conflict graph, which is a non-trivial extension of the counterpart in WDM networks. Several constant approximation ratios of RSA algorithm have also been derived through the theoretical analysis.
- By leveraging random graph theory, we provide an analytical approach on how to connect the chromatic number of conflict graph with the intersecting probability of the lightpath routing. Meanwhile, a matrix of conflict coefficients and the GOF, which embody the impact of the traffic distribution and network topology, are also proposed to determine the intersecting probability.
- Within the proposed theoretical analysis, we evaluated three realistic EONs under two traffic distributions by the conflict coefficients and GOFs. Extensive simulations have also been conducted, results of which verify the effectiveness of our

theoretical deductions.

The remaining of this chapter is organized as follows. We present the formulation of RSA problem and the conflict graph in Section 2.2. Then, the theoretical work that reveals the relation between the optimality of the lightpath routing and the chromatic number of its conflict graph is discussed in Section 2.3. In Section 2.4, we introduce the connection between intersecting probability and chromatic number, and propose the conflict coefficients and a quadratic programming. Within the proposed theoretical analysis, we evaluated three EONs under two traffic distributions in Section 2.5. Extensive simulations under different scenarios are conducted in Section 2.6 to verify our analysis. Finally, Section 2.7 summarizes this paper.

2.2 Network Model and Problem Description

In this section, we present the network model considered in this chapter and the formulation of RSA. Some necessary notations are summarized in Table 2.1.

2.2.1 Network Model

Network Topology

We use a directed graph $G(V, E)$ to represent the topology of an EON, where V and E denote the sets of nodes and directed fiber links respectively as in Fig. 2.1. A bunch of FSs lies on each directed fiber link as in Fig. 2.2.

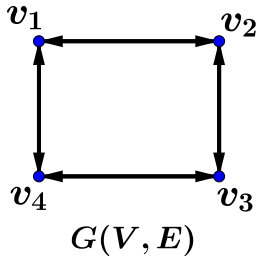


Figure 2.1: *An EON of 4 nodes and 4 bidirectional fiber links.*

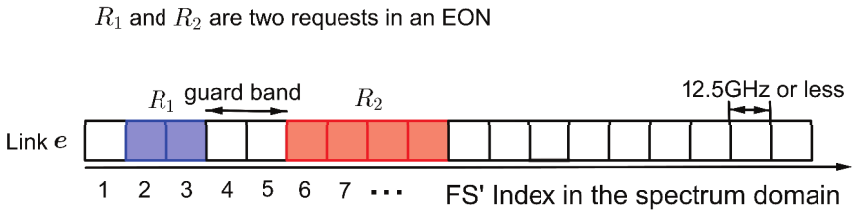


Figure 2.2: *FSs and guard-bands in a directed fiber link of an EON.*

Table 2.1: *Notations I of Chapter 2*

$G(V, E)$	The underlying EON, where V is the set of nodes, and E is the set of directed fiber links.
$\mathcal{D}$	The traffic distribution, which specially refers to the distribution of occurrence probabilities of source-destination pairs.
w_{sd}	The occurrence probability of source-destination pair (s, d) determined by the traffic distribution $\mathcal{D}$.
$\mathcal{R}$	The set of connection requests in $G(V, E)$.
n	$= \lvert\mathcal{R}\rvert$, the number of connection requests.
$R_i(s_i, d_i)$	$R_i \in \mathcal{R}$ represents the i-th connection request, where $s_i, d_i \in V$ are the source and destination nodes respectively generated by $\mathcal{D}$.
$\mathbb{N}^+$	The set of positive natural numbers representing the FS index set in the spectrum domain lying in each directed fiber link $e \in E$.
α	$\alpha \in \mathbb{N}^+$ is the upper bound of the bandwidth demanded by requests, expressed in the number of FSs.
β	$\beta \in \mathbb{N}^+$ is the lower bound of the bandwidth demanded by requests, expressed in the number of FSs.
R_i^w	The number of contiguous FSs (bandwidth requirement) required by R_i, which is in the range of $[\alpha, \beta]$.
$\mathcal{P}_i$	The set of all the directed paths from s_i to d_i in $G(V, E)$.
P_i	$P_i \in \mathcal{P}_i$ is the directed path on which R_i is routed.
W_i	The set of contiguous FSs assigned to R_i.
R_i^b	$R_i^b \in \mathbb{N}^+$ is the start-index of W_i.
R_i^a	$R_i^a \in \mathbb{N}^+$ is the end-index of W_i.
GB	$GB \in \mathbb{N}^+$ is the number of FSs required for the guard-band.
MUFI	$= \max_{s \in (\cup W_i)} (s)$, is the maximum used FS index.
$\hat{G}(\hat{V}, \hat{E})$	The conflict graph which is a weighted undirected graph, where $\hat{V}$ is the vertex set corresponding to $\mathcal{R}$, and $\hat{E}$ is the edge set.
$\hat{v}_i$	$\hat{v}_i \in \hat{V}$ corresponds to R_i.
$\hat{v}_i^w$	$= R_i^w$, the vertex weight of $\hat{v}_i$.
$W_{\hat{v}_i}$	The set of contiguous FSs assigned to $\hat{v}_i$.
$\hat{v}_i^b$	$\hat{v}_i^b \in \mathbb{N}^+$ is the start-index of $W_{\hat{v}_i}$.
$\hat{v}_i^a$	$\hat{v}_i^a \in \mathbb{N}^+$ is the end-index of $W_{\hat{v}_i}$.
$\hat{v}_{I(i)}^w$	The i-th biggest vertex weight in $\hat{V}$ *i.e.*, $\hat{v}_{I(1)}^w \geq \hat{v}_{I(2)}^w \geq ... \geq \hat{v}_{I(n)}^w$.
$\hat{v}_{D(i)}^w$	The i-th smallest vertex weight in $\hat{V}$ *i.e.*, $\hat{v}_{D(1)}^w \leq \hat{v}_{D(2)}^w \leq ... \leq \hat{v}_{D(n)}^w$.
$opt(\hat{G})$	The optimal spectrum assignment for $\hat{G}(\hat{V}, \hat{E})$.
$\lvert opt(\hat{G})\rvert$	The MUFI of $opt(\hat{G})$, which is the optimal one.
$\chi(\hat{G})$	The chromatic number of $\hat{G}$.

Traffic Distribution

In this chapter, the traffic distribution $\mathcal{D}$ specially refers to the distribution of occurrence probabilities of source-destination pairs. Taking $G(V, E)$ in Fig. 2.1 for example, if

the occurrence probabilities of the source-destination pairs (v_1, v_3) and (v_3, v_1) are 45% respectively, and the other node pairs equally take the remaining 10%, then the traffic distribution $\mathcal{D}$ is shown in Table 2.2.

Table 2.2: *Traffic Distribution $\mathcal{D}$*

Node Pair (s, d)	Occurrence Probability	Node Pair (s, d)	Occurrence Probability
(v_1, v_2)	$w_{v_1v_2} = 1\%$	(v_2, v_1)	$w_{v_2v_1} = 1\%$
(v_1, v_3)	$w_{v_1v_3} = 45\%$	(v_3, v_1)	$w_{v_3v_1} = 45\%$
(v_1, v_4)	$w_{v_1v_4} = 1\%$	(v_4, v_1)	$w_{v_4v_1} = 1\%$
(v_2, v_3)	$w_{v_2v_3} = 1\%$	(v_3, v_2)	$w_{v_3v_2} = 1\%$
(v_2, v_4)	$w_{v_2v_4} = 1\%$	(v_4, v_2)	$w_{v_4v_2} = 1\%$
(v_3, v_4)	$w_{v_3v_4} = 1\%$	(v_4, v_3)	$w_{v_4v_3} = 1\%$

2.2.2 Formulation of RSA

In this thesis, we study the RSA problem with a planning concern, *i.e.*, given a set of connection requests $\mathcal{R}$ in an EON $G(V, E)$, we intend to minimize the spectrum usage in optical fibers. For each request $R_i(s_i, d_i) \in \mathcal{R}$, we need to establish a lightpath and assign enough bandwidth on it so as to forward the data of the request. More specially, the RSA problem consists in choosing a path P_i from the set $\mathcal{P}_i$ for R_i and assigning just enough FS set W_i on this path while satisfying the following constraints:

- **Bandwidth Requirement Constraint**. The number of FSs assigned to each request should no smaller than its bandwidth requirement, *i.e.*, the cardinality of W_i assigned to R_i must be equal to its weight:

$$|W_i| = R_i^w, \quad \forall R_i \in \mathcal{R}. \tag{2.1}$$

  Without loss of generality, we make the following assumption in this chapter.
  Hypothesis 2.2.1. *The bandwidth of each request is uniformly distributed in the range of $[\alpha, \beta]$, i.e., $R_i^w \in [\alpha, \beta]$ $\forall i$, where α and β are two constant integers, e.g., $[\alpha, \beta] = [1, 2]$ in [71] or $[1, 3]$ in [72].*

- **Spectrum Contiguity Constraint**. The FSs assigned to request R_i must be contiguous in the spectrum domain, *i.e.*, $\mathbb{N}^+$. Thus, W_i can be expressed as $\{R_i^b, R_i^b + 1, ..., R_i^a - 1, R_i^a\}$. This is a physical layer constraint for all-optical communications.

- **Spectrum Continuity Constraint**. All the directed fiber links on the lightpath for R_i (*i.e.*, $e \in P_i$) should be assigned with the same set of contiguous FSs W_i.

- **Guard Band Constraint**. To mitigate mutual interference, when P_i and P_j share common some directed fiber link(s), the distance between W_i and W_j in the spectrum domain should be no less than GB (as shown in Fig. 2.2):

$$distance(W_i, W_j) \geq GB, \quad \forall P_i \cap P_j \neq \emptyset, \tag{2.2}$$

where,

$$distance(W_i, W_j) = \min_{s \in W_i, t \in W_j} (|s - t| - 1).$$

Note that in this chapter, for ease of the study of the sophisticated topic, the size of GB is set as a constant number of FSs for all request pairs whose lightpaths intersect.

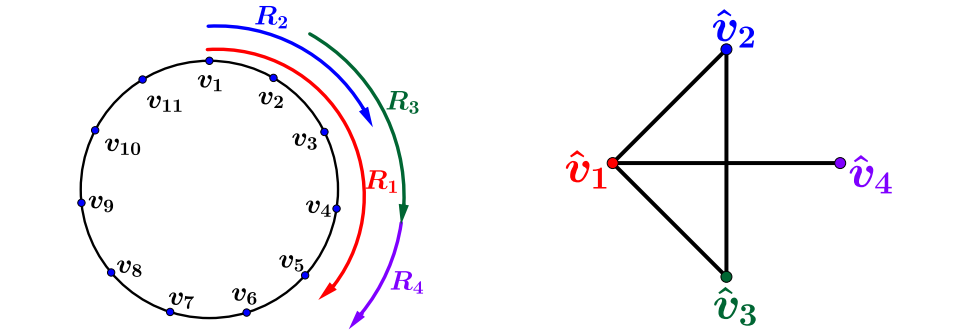
For planning purposes, the RSA problem studied in this thesis aims to minimize the **Maximum Used FS Index** (MUFI), which can be expressed by Eq. (2.3):

$$\min \left[\max_{s \in (\cup W_i)} (s) \right] \qquad (\textbf{RSA}). \tag{2.3}$$

2.2.3 Conflict Graph

To solve the RSA problem, we should first compute a lightpath for each request, and then minimize the MUFI. Since the latter is similar to the classic coloring problems to some extent [75], a practical solution is to build an auxiliary graph, namely the conflict graph, whose formal definition is given as follows.

Definition 2.2.1. *The conflict graph [78]* $\hat{G}(\hat{V}, \hat{E})$ [1] *is such* ***a weighted undirected graph*** *whose vertex set* $\hat{V}$ *represents the set of requests, i.e.,* $\mathcal{R}$*. Any two vertices* $\hat{v}_i, \hat{v}_j \in \hat{V}$ *(representing* R_i *and* R_j *respectively) are connected by an edge* $\hat{e} \in \hat{E}$*, i.e., they are adjacent in* $\hat{G}$*, if and only if* P_i ***intersects with*** P_j*, i.e.,* $P_i \cap P_j \neq \emptyset$ ***(at least one directed fiber link shared by*** P_i ***and*** P_j***)****, where* P_i *and* P_j *are the lightpaths for* R_i *and* R_j *respectively. We denote by* $\hat{v}_i^w$ *the weight of vertex* $\hat{v}_i$*, and* $\hat{v}_i^w = R_i^w$*. Besides,* $\hat{v}_i^b$*,* $\hat{v}_i^a$ *and* $W_{\hat{v}_i}$ *have the same meanings as* R_i^b*,* R_i^a *and* W_i *respectively. If* $\hat{v}_i$ *and* $\hat{v}_j$ *are adjacent in* $\hat{G}$*, then the distance between* $W_{\hat{v}_i}$ *and* $W_{\hat{v}_j}$ *should be no less than* GB*.*



(a) Routed lightpaths. (b) Conflict graph.

Figure 2.3: *The conflict graph of the routed lightpaths.*

[1] In this chapter, to distinguish the conflict graph from the EON, we use $\hat{G}$ to represent a conflict graph.

Figure 2.3(b) showcases a 4-node conflict graph for the 4 requests (they are already routed on their lightpaths) in Fig. 2.3(a), where $\hat{v}_i$ corresponds to $R_i, \forall 1 \leq i \leq 4$. According to the definition, obviously, any proper spectrum assignment for the conflict graph corresponds to a proper spectrum assignment for the requests, vice versa. Thus, the conflict graph embodies all the four constraints of RSA mentioned above, and is very important to analyze the RSA problem. Given a set of requests $\mathcal{R}$, no matter how they are routed, we can use the conflict graph to characterize the intersections among them.

2.3 Optimal MUFI and Chromatic Number of Conflict Graph

Since the conflict graph utterly depicts the lightpath routing, it is important to characterize its graph property so as to study its impact on the final MUFI.

Obviously, once a conflict graph $\hat{G}$ is established, there is an optimal spectrum assignment $opt(\hat{G})$ which produces the minimum MUFI, *i.e.*, the optimal MUFI denoted by $|opt(\hat{G})|$. Conversely, the optimal MUFI $|opt(\hat{G})|$ can be viewed as the lower bound on $\hat{G}$ that all feasible spectrum assignments can reach. Thus, the optimal MUFI $|opt(\hat{G})|$ of a conflict graph $\hat{G}$ is vital to the final performance of the RSA problem. Next, we first explore the relation between the $|opt(\hat{G})|$ and the conflict graph $\hat{G}$ itself.

2.3.1 Optimal MUFI and Chromatic Number

In WDM networks, the parallel relation is that the minimum number of wavelengths used on a conflict graph is equal to its chromatic number. However, in EONs, how to determine the $|opt(\hat{G})|$ for a conflict graph $\hat{G}$ has not been investigated yet. Thus, we address this issue by giving Theorem 2.3.1.

Theorem 2.3.1. *Assuming $\hat{G}(\hat{V}, \hat{E})$ is the conflict graph, then*

$$(\chi(\hat{G}) - 1) \cdot GB + \sum_{i=1}^{\chi(\hat{G})} \hat{v}^w_{D(i)} \leq |opt(\hat{G})| \leq (\chi(\hat{G}) - 1) \cdot GB + \sum_{i=1}^{\chi(\hat{G})} \hat{v}^w_{I(i)}. \tag{2.4}$$

where $\hat{v}^w_{I(i)}$ and $\hat{v}^w_{D(i)}$ are the i-biggest and i-smallest vertex weight in $\hat{V}$ respectively, and $\chi(\hat{G})$ is the chromatic number of $\hat{G}$.

Proof. We prove the upper bound by finding a feasible spectrum assignment f whose MUFI is smaller than it. This solution f can be obtained in such way as shown in Fig. 2.4: (1) separating $\hat{V}$ into $\chi(\hat{G})$ disjoint independent sets; (2) assigning FSs for each independent set (the number of FSs assigned is equal to the biggest vertex weight in this set); (3) patching them up. It is easy to see the MUFI of f is smaller than the upper bound.

Now, we prove the lower bound. Let $opt(\hat{G}) = \{W_{\hat{v}}, \forall \hat{v} \in \hat{V}\}$ be the optimal spectrum assignment. With respect to $opt(\hat{G})$, let $A = \{\hat{v}^a, \forall \hat{v} \in \hat{V}\}$ and $B = \{\hat{v}^b, \forall \hat{v} \in \hat{V}\}$

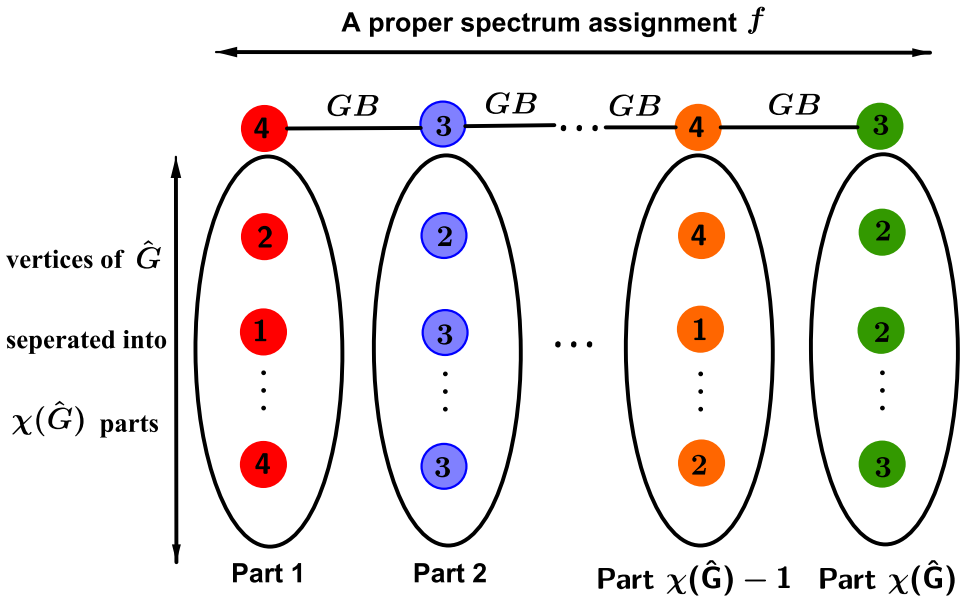


Figure 2.4: *A feasible spectrum assignment f where each circle represents a vertex and the number in the circle is the corresponding vertex weight.*

be the optimal end-index and start-index sets for all $W_{\hat{v}}$ in $opt(\hat{G})$ respectively. We then can separate $\hat{V}$ into different parts as follows:

First, let, *w.l.o.g.*, $\hat{v}_1$ be the vertex whose end-index $\hat{v}_1^a$ is the minimum in A, then we assert that $\mathcal{F}_1 = \{\hat{v}|\hat{v}^b \leq \hat{v}_1^a + GB, \forall \hat{v} \in \hat{V}\}$ is an independent set of $\hat{G}$, *i.e.*, for any two vertices $\hat{v}_i, \hat{v}_j \in \mathcal{F}_1$, $\hat{v}_i$ is not adjacent to $\hat{v}_j$ in $\hat{G}$. We prove it by contradiction. Assuming $\hat{v}_i$ is adjacent to $\hat{v}_j$ in $\hat{G}$ and $W_{\hat{v}_i} = [\hat{v}_i^b, \hat{v}_i^a]$ and $W_{\hat{v}_j} = [\hat{v}_j^b, \hat{v}_j^a]$, we have $distance(W_{\hat{v}_i}, W_{\hat{v}_j}) \geq GB$. Thus, we, *w.l.o.g.*, can assume $\hat{v}_i^a + GB < \hat{v}_j^b$. However, according to $\mathcal{F}_1$, we have $\hat{v}_j^b \leq \hat{v}_1^a + GB$ and $\hat{v}_i^a \geq \hat{v}_1^a$ (since $\hat{v}_1^a$ is the minimum in all end-indices), which means $\hat{v}_j^b \leq \hat{v}_i^a + GB$, a contradiction. Therefore $\mathcal{F}_1$ is an independent set of $\hat{G}$.

Next, let, *w.l.o.g.*, $\hat{v}_2$ be the vertex whose end-index is the minimum in $\hat{V} \setminus \mathcal{F}_1$, and similarly, we can assert $\mathcal{F}_2 = \{\hat{v}|\hat{v}^b \leq \hat{v}_2^a + GB, \forall \hat{v} \in \hat{V} \setminus \mathcal{F}_1\}$ is an independent set of $\hat{G}$. After finite steps using the same technique, say k, we can separate $\hat{V}$ into k disjoint independent sets: $\mathcal{F}_1$, $\mathcal{F}_2$,..., $\mathcal{F}_k$. Besides, according to the principle of selecting $\mathcal{F}_i, \forall 1 \leq i \leq k$, we have $\hat{v}_i^a + GB < \hat{v}_{i+1}^b, \forall 1 \leq i \leq k$. Therefore, we have $(k-1) \cdot GB + \sum_{i=1}^{k} \hat{v}_i^w \leq |opt(\hat{G})|$. Figure 2.5 sketches the process above.

Finally, according to the definition of chromatic number that $\chi(G)$ is the minimum number of independent sets into which we can separate $\hat{V}$, so we immediately have $\chi(G) \leq k$. Besides $\sum_{i=1}^{k} \hat{v}_{D(i)}^w \leq \sum_{i=1}^{k} \hat{v}_i^w$, we hence have $(\chi(\hat{G})-1) \cdot GB + \sum_{i=1}^{\chi(\hat{G})} \hat{v}_{D(i)}^w \leq$

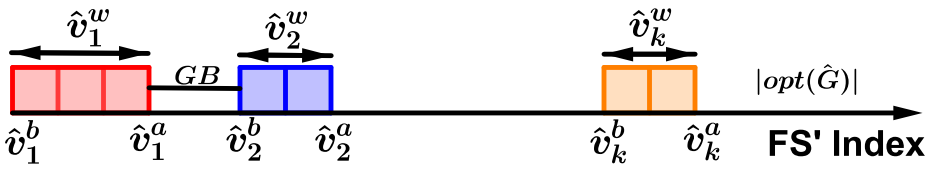


Figure 2.5: *The lower bound of* $|opt(\hat{G})|$.

$(k-1)\cdot GB + \sum_{i=1}^{k} \hat{v}^w_{D(i)} \leq (k-1)\cdot GB + \sum_{i=1}^{k} \hat{v}^w_i \leq |opt(\hat{G})|$. Thus, we get the proof. □

Theorem 2.3.1 reveals the relation between the chromatic number of the conflict graph $\hat{G}$ and the optimal MUFI $|opt(\hat{G})|$, which is a non-trivial generation from RWA. Under Hypothesis 2.2.1 and Theorem 2.3.1, we can get Corollary 2.3.1.

Corollary 2.3.1. *Assuming* $\hat{G}(\hat{V},\hat{E})$ *is the conflict graph and* $\hat{v}^w \in [\alpha,\beta]$, *then we have*

$$(\chi(\hat{G})-1)\cdot GB + \chi(\hat{G})\cdot\alpha \leq |opt(\hat{G})| \leq (\chi(\hat{G})-1)\cdot GB + \chi(\hat{G})\cdot\beta. \tag{2.5}$$

In practice, as $\hat{v}^w_{D(i)}$ and $\hat{v}^w_{I(n-i)}$ will become closer with the growth of i, the gap between the upper and lower bounds in Theorem 2.3.1 will be smaller than that in Corollary 2.3.1. In other words, the $|opt(\hat{G})|$ is limited in a narrow interval bounded by the chromatic number $\chi(\hat{G})$. Thus, $\chi(\hat{G})$ directly determines $|opt(\hat{G})|$.

2.3.2 Some Approximation Results

After the lightpath routing, the conflict graph $\hat{G}(\hat{V},\hat{E})$ is constructed. Now, a natural question is how to obtain a good spectrum assignment to approach $|opt(\hat{G})|$. Since the $|opt(\hat{G})|$ is bounded by the chromatic number $\chi(\hat{G})$, we can get an approximation algorithm (say APX) of spectrum assignment from a graph coloring algorithm (say COL) by slightly modifying it as follows: (1) using COL to partition the vertex set $\hat{V}$ into k disjoint independent sets (each set in a monochrome), *i.e.*, C_1, C_2,...,C_k, where k is the chromatic number produced by COL; (2) using the same way that we obtain the upper bound in Theorem 2.3.1 (in the Appendix A) to assign FS sets. We get Corollary 2.3.2.

Corollary 2.3.2. *Given a conflict graph* $\hat{G}(\hat{V},\hat{E})$, *if there is a polynomial algorithm COL which can guarantee a* ρ *approximation ratio for chromatic number, then there is a polynomial algorithm APX which can guarantee a* $\rho\cdot max\{\frac{\beta}{\alpha},2\}$ *approximation ratio for* $|opt(\hat{G})|$, *where* α *and* β *are defined in Hypothesis 2.2.1.*

Proof. Since $\chi(\hat{G}) = 1$ is a trivial case, we just consider $\chi(\hat{G}) \geq 2$. Supposing that COL separates $\hat{V}$ into $C_1, C_2, ..., C_k$, let, *w.l.o.g.*, $\hat{v}_1, \hat{v}_2, ..., \hat{v}_k$ be the vertices with the biggest vertex weight in each independent set respectively (just like that in Fig. 2.4). Then according to the construction of APX, we have $|APX(\hat{G})| \leq (k-1)\cdot GB + \sum_{i=1}^{k} \hat{v}^w_i$,

where $|APX(\hat{G})|$ is the MUFI computed by APX. According to Corollary 2.3.1 and the deduction above, we have $\frac{|APX(\hat{G})|}{|opt(\hat{G})|} \leq \frac{(k-1)\cdot GB + k\cdot\beta}{(\chi(\hat{G})-1)\cdot GB + \chi(\hat{G})\cdot\alpha} \leq \max\{\rho\cdot\frac{\beta}{\alpha}, \frac{k-1}{\chi(\hat{G})-1}\}$. Since $\chi(\hat{G}) \geq 2$, $\frac{k-1}{\chi(\hat{G})-1} \leq 2\cdot\frac{k}{\chi(\hat{G})} = 2\cdot\rho$. Hence, $\frac{|APX(\hat{G})|}{|opt(\hat{G})|} \leq \rho\cdot\max\{\frac{\beta}{\alpha}, 2\}$. □

For some special conflict graphs, their chromatic number can be solved in polynomial time. Here, we have Corollary 2.3.3.

Corollary 2.3.3. *If conflict graph $\hat{G}(\hat{V},\hat{E})$ is a perfect graph, then there is a polynomial algorithm which can guarantee a $\frac{\beta}{\alpha}$ approximation ratio for $|opt(\hat{G})|$.*

Proof. According to [31], the chromatic numbers of perfect graphs can be solved in polynomial time, *i.e.*, there is an algorithm such that $\rho = 1$. In the proof of Corollary 2.3.2, we have $\frac{|APX(\hat{G})|}{|opt(\hat{G})|} \leq \max\{\rho\cdot\frac{\beta}{\alpha}, \frac{k-1}{\chi(\hat{G})-1}\}$. By $\rho = \frac{k}{\chi(\hat{G})} = 1$, and $\frac{\beta}{\alpha} \geq 1$, we get this Corollary. □

In EONs with a tree topology, the conflict graphs over them are always perfect graphs. Then, we have:

Corollary 2.3.4. *If EON $G(V,E)$ is a tree, there is a polynomial algorithm which can guarantee a $\frac{\beta}{\alpha}$ ratio for the optimal MUFI.*

Proof. Because the $G(V,E)$ is a tree, there is only one way to route each request, *i.e.*, the conflict graph is unique. Besides, according to [27], the conflict graphs over a tree network are always chordal graphs which belong to the perfect graph class. Therefore, the proof follows. □

However, in general cases, for a conflict graph $\hat{G}(\hat{V},\hat{E})$, it is intractable to approximate $\chi(\hat{G})$ within a ratio of $|\hat{V}|^{1-\epsilon}$ for any constant $\epsilon > 0$ [24], *i.e.*, compared with $\chi(\hat{G})$, the final result could be very bad for any polynomial-time algorithm. Now, the best approximation ratio ρ for $\chi(\hat{G})$ is $\mathcal{O}(|\hat{V}|\cdot(\log\log|\hat{V}|)^2/\log(|\hat{V}|)^3)$ [32].

From above discussion, we can see that the spectrum assignment, closely analogous to the graph coloring problem, is intractable, and the $|opt(\hat{G})|$ is determined by the conflict graph of the routed lightpahs. Therefore, a good routing scheme should be to reduce the chromatic number $\chi(\hat{G})$, thereby the $|opt(\hat{G})|$, which is critical for the final performance of RSA.

2.4 Theoretical Chains of the Impact

In the previous section, we theoretically deduced the impact of $\chi(\hat{G})$ on the $|opt(\hat{G})|$, *i.e.*, the minimum MUFI that $\hat{G}$ can obtain. Now, the important thing is to study how the traffic distribution $\mathcal{D}$ and the EON topology G affect $\chi(\hat{G})$. Once we figure this out, the impact of these two factors on $|opt(\hat{G})|$ will become clear.

To this end, we need some preparations in random graph theory. We first introduce the concept of intersecting probability and its relation with the chromatic number. Then, we shall see how the intersecting probability can be influenced by the traffic distribution $\mathcal{D}$ and network topology G. Thus, by the connection of intersecting probability, we finally figure out the impacts of the two factors. Here, we need some other necessary notations summarized in Table 2.3.

Table 2.3: *Notations II of Chapter 2*

p	The intersecting probability of the lightpaths for any two requests.
K	The number of candidate paths for each source-destination pair.
p_i	The probability for a request to be routed on the i-th candidate path.
θ_{ij}	The conflict coefficient that represents the intersecting probability of any two requests, one routed on the i-th candidate path and the other on the j-th path.
CM	$= [\theta_{ij}]$, the real symmetric Conflict Matrix composed of all θ_{ij}.

2.4.1 Intersecting Probability and Chromatic Number

In a backbone EON $G(V, E)$, the traffic distribution $\mathcal{D}$ may statistically has some characteristics: For example, each source-destination pair in G may occur with equiprobability, or there are some important date-center nodes which constitute the majority of the connection requests. Support we have already at hand both the traffic distribution $\mathcal{D}$ and the EON topology $G(V, E)$.

Under the random circumstance, the source and the destination of each request are generated by $\mathcal{D}$, and its bandwidth follows a uniform distribution in the range $[\alpha, \beta]$ by Hypothesis 2.2.1. In light of this, each request R_i can thus be viewed as a random variable. Herein, we introduce the intersecting probability of the lightpaths for any two requests.

Definition 2.4.1. *In an EON $G(V, E)$, the intersecting probability (denoted by p) of any two requests generated by $\mathcal{D}$ (say R_1 and R_2) is the probability that their lightpaths P_1 and P_2 share at least one common fiber link, i.e., $P_1 \cap P_2 \neq \emptyset$.*

Given $\mathcal{R} = \{R_1, R_2, ..., R_n\}$, the vertex set of the conflict graph $\hat{G}(\hat{V}, \hat{E})$ is that $\hat{V} = \{\hat{v}_1, \hat{v}_2, ..., \hat{v}_n\}$. Thus, the edge set $\hat{E}$ determines the final $\chi(\hat{G})$. For any two vertices, say $\hat{v}_1, \hat{v}_2 \in \hat{V}$, the probability that they are adjacent in the conflict graph $\hat{G}$ is equal to the intersecting probability p. We introduce an important Lemma in random graph theory which reveals the relation between the probability that two vertices are adjacent in $\hat{G}$ and the chromatic number $\chi(\hat{G})$.

Lemma 2.4.1. *[12] Let $|\hat{V}| = n$, p be the intersecting probability for any two vertices in $\hat{V}$, and $\hat{G}$ be the conflict graph, then we have $\chi(\hat{G}) = (\frac{1}{2} + o(1)) \cdot \log(\frac{1}{1-p}) \cdot \frac{n}{\log n}$, where $o(1)$ is an infinitesimal of n.*

From Lemma 2.4.1, we can see that there is a strongly positive correlation between the $\chi(\hat{G})$ and the intersection probability p. More specifically, the smaller the intersection probability p, the smaller the $\chi(\hat{G})$, which conforms to our intuition. Meanwhile, by Theorem 2.3.1, the smaller chromatic number $\chi(\hat{G})$, the smaller $|opt(\hat{G})|$. Therefore, we obtained the first important theoretical chain in Fig. 2.6.

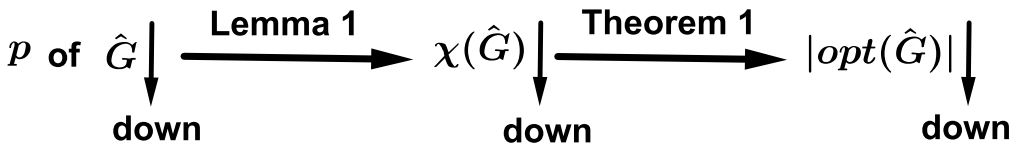
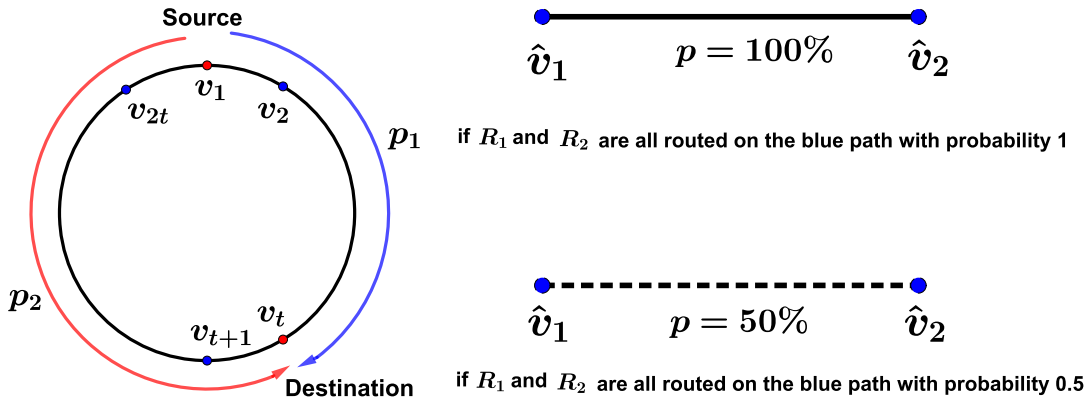


Figure 2.6: *Theoretical Chain 1.*

Now, the question is that how to determine the intersecting probability p of the lightpath routing. Obviously, the p depends on three factors: ① the traffic distribution $\mathcal{D}$, ② the EON topology $G(V, E)$, and ③ the routing scheme. We use a simple example below to illustrate that.



(a) Network Configuration. (b) Intersecting Probabilities.

Figure 2.7: *An example of intersecting probability.*

Example 2.4.1. *The network topology of the EON is a bidirectional cycle $G(V, E) = [v_1 v_2 ... v_{2t} v_1]$, whose vertice are labeled clockwise from v_1 to v_{2t} as shown in Fig. 2.7. We assume the following traffic distribution $\mathcal{D}$: The occurrence probability of the source-destination pair (v_1, v_t) is $w_{v_1 v_t} = 100\%$ while it is zero for the others. We can see there are two candidate paths marked by blue and red respectively.*

If the routing scheme routes all the requests on the first shortest path (the blue one), then the lightpath intersecting probability p is 100%, which will produce a complete conflict graph $\hat{G}$ with the biggest chromatic number, thereby resulting in the maximum

$|opt(\hat{G})|$. Obviously, it is a bad routing scheme. An intuitively optimal routing scheme is that half of the requests are routed on the red one and the remaining on the blue one. In other words, the request (since we treat each request as a random variable in this chapter) should be routed on the shortest path with probability 0.5 and the second path with the same probability. With this new routing scheme, the intersecting probability becomes $p = 50\%$.

From the above example, it is not difficult to observe that the routing scheme should take into account the traffic distribution $\mathcal{D}$ and the EON topology. Besides, we can also find that there is a lower bound on the intersecting probability p. In other words, no matter what routing scheme is used, there is a minimum intersecting probability p_{min} that we can not decrease, for instance $p_{min} = 50\%$ for the above example. The limitation on p_{min} obviously comes from the impact of $\mathcal{D}$ and G. Next, we investigate the mechanism of their impact.

2.4.2 Conflict Coefficients

Here, we introduce a new concept named **conflict coefficients** of an EON, which will play an important role for lightpath routing in EONs.

Definition 2.4.2. *The conflict coefficient θ_{ij} of EON $G(V, E)$ is the intersecting probability of any two requests, say $R_1(s_1, d_1)$ and $R_2(s_2, d_2)$ generated by $\mathcal{D}$, which are routed on the i-th shortest path of (s_1, d_1) and j-th shortest path of (s_2, d_2) in G respectively.*

Table 2.4: *Matrix $M^{\theta_{ij}}$ for the conflict coefficient θ_{ij}.*

		On the j-th path					
		.	.	(s_2, d_2) $w_{s_2 d_2}$	.	.	
On	.	.	.	.	.	.	
the	(s_1, d_1)	.	.	$w_{s_1 d_1} \times w_{s_2 d_2}$	.	.	
i-th	$w_{s_1 d_1}$	.	.	.	.	.	
path	.	.	.	.	.	.	
		$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{|V|\times(|V|-1)}$					

The θ_{ij} is related to $\mathcal{D}$ and $G(V, E)$, which can be computed as follows. We first generate the i-th and j-th shortest paths for all $|V| \times (|V| - 1)$ source-destination pairs in $G(V, E)$. We construct a matrix $M^{\theta_{ij}}$ in Table 2.4.

The top row in Table 2.4 represents that all the $|V| \times (|V| - 1)$ source-destination pairs are routed on their own j-th shortest paths, while the leftmost column means the $|V| \times (|V| - 1)$ source-destination pairs are routed on their own i-th shortest paths (Note that requests with the same s and d are regarded as different). The weight w_{sd} for each source-destination pair (s, d) is its occurrence probability determined by the traffic distribution $\mathcal{D}$. If the i-th shortest path of (s_1, d_1) intersects with the j-th shortest

path of (s_2, d_2) in $G(V, E)$, then this entry is $w_{s_1 d_1} \times w_{s_2 d_2}$, otherwise 0. Then, we have Theorem 2.4.1.

Theorem 2.4.1. *The conflict coefficient $\theta_{ij} = \sum M^{\theta_{ij}}$, where $\sum M^{\theta_{ij}}$ represents the sum of all entries in the matrix.*

Proof. According to the definition, θ_{ij} is the intersecting probability of any two requests, say R_1 routed on the i-th shortest path and R_2 on the j-th shortest path. Since both R_1 and R_2 are generated by the traffic distribution $\mathcal{D}$ in $G(V, E)$, the probability that the source-destination pairs of R_1 and R_2 are (s_1, d_1) and (s_2, d_2) respectively is $w_{s_1 d_1} \times w_{s_2 d_2}$. Therefore, $\theta_{ij} = \sum_{\left((s_1,d_1)^i \cap (s_2,d_2)^j \neq \emptyset\right)} w_{s_1 d_1} \times w_{s_2 d_2}$, where $(s_1, d_1)^i \cap (s_2, d_2)^j \neq \emptyset$ means that the i-th shortest path of (s_1, d_1) intersects with the j-th shortest path of (s_2, d_2) in the $G(V, E)$. Finally, it is easy to see that $\sum_{\left((s_1,d_1)^i \cap (s_2,d_2)^j \neq \emptyset\right)} w_{s_1 d_1} \times w_{s_2 d_2} = \sum M^{\theta_{ij}}$ and the proof follows. □

The conflict coefficient θ_{ij} is an integral reflection on the impact of both the traffic distribution $\mathcal{D}$ and the EON topology G. Following the same idea, we can compute all the conflict coefficients of an EON G under a specific traffic distribution $\mathcal{D}$, which compose **a real symmetric Conflict Matrix (CM)** as below. CM is an important evaluation metric for an EON G under a specific traffic distribution $\mathcal{D}$ as we shall see later.

$$CM = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \dots & \theta_{1K} & \dots \\ \theta_{21} & \theta_{22} & \theta_{23} & \dots & \theta_{2K} & \dots \\ & \dots\dots\dots\dots & & & & \\ \theta_{K1} & \theta_{K2} & \theta_{K3} & \dots & \theta_{KK} & \dots \\ & \dots\dots\dots\dots & & & & \end{bmatrix}$$

2.4.3 Global Optimal Formulation (GOF)

In general, the number of paths connecting a source-destination pair is exponential in an EON. It is impossible to enumerate all possible paths. Thus, a practical way is to precompute a set of K shortest candidate paths for each source-destination pair by some K-shortest path algorithms, and then route each request on one of them. Given a set of $\mathcal{R} = \{R_1, R_2, ..., R_n\}$, we use p_i to denote the percentage of requests which are routed on the i-th shortest candidate path and $\sum_{i=1}^{K} p_i = 1$. The proportion array $(p_1, p_2, ..., p_k)$ is determined by the routing scheme used. As mentioned above that each request is treated as a random variable independently generated by $\mathcal{D}$ in this chapter, p_i can also be interpreted as the probability of the request routed on the i-th shortest candidate path.

Thus, for any two requests, say R_1 and R_2, the probability that the R_1 routed on the i-th shortest candidate path and R_2 on the j-th shortest candidate path is $p_i p_j$. Combing

with conflict coefficient θ_{ij}, the conditional probability that any two requests intersect, routed on i-th and j-th shortest candidate paths respectively, is $\theta_{ij}p_ip_j$. Finally, we sum all the conditional probabilities and obtain a Global Optimal Formulation (GOF) as follows, which is a quadratic programming, and determines the intersecting probability p:

$$p = \sum_{1\le i,j\le K} \theta_{ij}p_ip_j \quad \textbf{(GOF)}, \tag{2.6}$$

$$\textbf{s.t.} \quad \sum_{i=1}^{K} p_i = 1, \tag{2.7}$$

$$p_i \ge 0, \quad 1 \le i \le K. \tag{2.8}$$

Here, K is the number of pre-computed candidate paths per source-destination pair, and $\theta_{ij}, \forall i, j$ are the conflict coefficients. In GOF, K is predetermined, and θ_{ij} are the parameters determined by the traffic distribution $\mathcal{D}$ and the network topology G, while p_i is determined by the routing scheme used. Thus, the complexity of the quadratic programming is with one constraint, K variables and K^2 parameters. By now, we summarize what we have obtained in Fig. 2.8 which is another important theoretic chain.

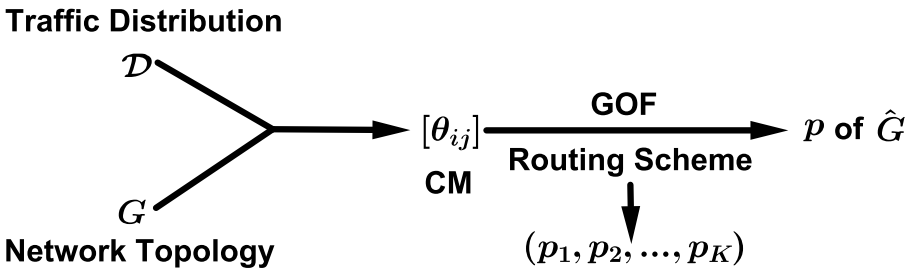


Figure 2.8: Theoretical Chain 2.

Combing Theoretic Chain 1 in Fig. 2.6 with Theoretic Chain 2 in Fig. 2.8, we finally figured out the impact of the traffic distribution $\mathcal{D}$ and network topology G on the lightpath routing. More specifically, the CM composed of the conflict coefficients, embodies the impact of the traffic distribution $\mathcal{D}$ and network topology G. All the conflict coefficients can be viewed as the capacity of an EON G under a specific traffic distribution $\mathcal{D}$, which determine the final performance of lightpath routing.

Besides, we can also see that the routing scheme, which determines the proportional array $(p_1, p_2, ..., p_K)$, is another important factor to determine the intersecting probability. A good routing scheme should get a proper array $(p_1, p_2, ..., p_K)$ resulting in a small intersecting probability p. In fact, after obtaining the CM, we can minimize the GOF to obtain the optimal $(p_1, p_2, ..., p_K)$ which reaches the minimum intersecting probability p. We also use Example 2.4.1 to illustrate this. In the Example 2.4.1 of Fig. 2.7, we set $K = 2$. According to the traffic distribution ($w_{v_1v_t} = 100\%$) and the topology of cycle-EON, the conflict coefficients are $\theta_{11} = 1$, $\theta_{12} = \theta_{21} = 0$ and $\theta_{22} = 1$. Therefore, we can

compute the intersecting probability p of any two requests in this example as follows.

$$\begin{aligned} & p = p_1^2 + p_2^2, \\ s.t. \quad & p_1 + p_2 = 1. \end{aligned} \tag{2.9}$$

Obviously, the intersecting probability p in Eq. (2.9) reaches the minimum when $p_1 = p_2 = 0.5$. It means the optimal routing scheme should route half of the requests on one candidate path and the remaining on the other one, which conforms to the intuition.

2.5 CM Estimation and Optimal Routing Decision in Realistic EONs

In this section, we estimate the conflict coefficients and solve the corresponding GOFs in three realistic EONs under two traffic distributions respectively. The three EONs are, as shown in Fig. 2.9, the Ring, NSFNET and NJ-LATA [18] respectively. The three EONs are of almost the same size in terms of the number of nodes. Thus, we also uniformly compare the three EONs from the perspective of intersecting probability. Since in the Ring, there are only two candidate paths for each source-destination pair, to make a fair comparison, we set $K = 2$ for all EONs by pre-computing the first and second shortest paths for each source-destination pair. Therefore, the intersecting probability p in GOF can be written as

$$p = \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

or

$$p = \theta_{11} \times p_1^2 + 2\theta_{12} \times p_1 p_2 + \theta_{22} \times p_2^2, \tag{2.10}$$

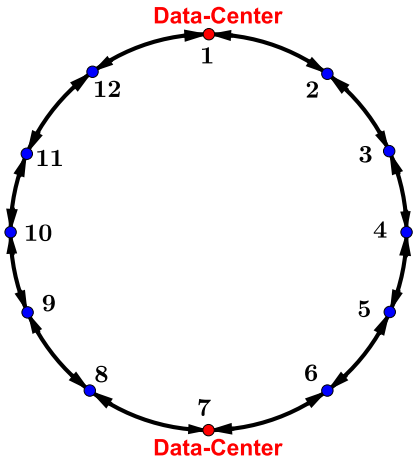
where, $p_1 + p_2 = 1$.

In the following, we compute the intersecting probability p under two traffic distribution scenarios respectively: uniform and weighted.

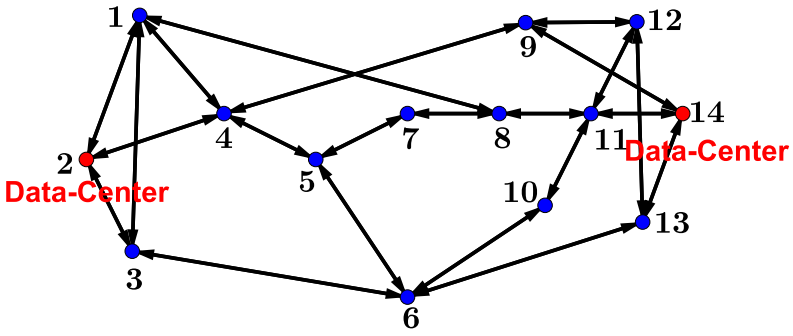
2.5.1 Uniform Traffic Distribution

In this subsection, we consider a uniform traffic distribution $\mathcal{D}$ in the three EONs, *i.e.*, each source-destination pair occurs with the same occurrence probability $\frac{1}{|V| \times (|V|-1)}$. Following the computing method in Section 2.4, we can get the corresponding conflict coefficients and GOFs of the three EONs under uniform distribution.

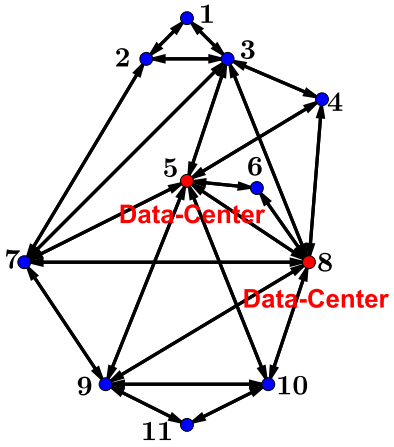
We denoted by CM_{Ring}^{Uni}, CM_{NSF}^{Uni} and CM_{NJ}^{Uni} the conflict matrices of the Ring, NSFNET and NJ-LATA respectively in the uniform traffic distribution. Following the computing method in Section 2.4.2, the three CMs are as follows.



(a) Ring (12 nodes and 12 bidirectional fiber links).



(b) NSFNET (14 nodes and 22 bidirectional fiber links).



(c) NJ-LATA (11 nodes and 23 bidirectional fiber links).

Figure 2.9: *Three realistic EON topologies.*

$$CM_{Ring}^{Uni} = \begin{bmatrix} 0.2328 & 0.4360 \\ 0.4360 & 0.5014 \end{bmatrix}, CM_{NSF}^{Uni} = \begin{bmatrix} 0.0979 & 0.1377 \\ 0.1377 & 0.2042 \end{bmatrix}, CM_{NJ}^{Uni} = \begin{bmatrix} 0.0901 & 0.0852 \\ 0.0852 & 0.1157 \end{bmatrix}.$$

Also we can obtain the the GOFs of the three EONs and the corresponding minimum intersecting probabilities in the uniform distribution as follows.

- For the Ring
$$p = 0.2328 \times p_1^2 + 0.8720 \times p_1 p_2 + 0.5014 \times p_2^2, \tag{2.11}$$
the minimum intersecting probability $p_{min} = 23.28\%$ when $p_1 = 1$ and $p_2 = 0$.
- For the NSFNET
$$p = 0.0979 \times p_1^2 + 0.2754 \times p_1 p_2 + 0.2042 \times p_2^2, \tag{2.12}$$
the minimum intersecting probability $p_{min} = 9.79\%$ when $p_1 = 1$ and $p_2 = 0$.
- For the NJ-LATA
$$p = 0.0901 \times p_1^2 + 0.1704 \times p_1 p_2 + 0.1157 \times p_2^2, \tag{2.13}$$
the minimum intersecting probability $p_{min} = 8.94\%$ when $p_1 = 0.8621$ and $p_2 = 0.1379$.

2.5.2 Weighted Traffic Distribution

Nowadays, EONs begin to support new networking capabilities and demanding network services such as data centers and clouds. Hence, the Optical Cross-Connect (OXC) in EONs connected to data-centers will produce a large amount of traffic among them, for instance data migration and content provisioning. This kind of traffic contributes to the majority of the total traffic. In other words, the nodes connected to data centers tend to have a much higher occurrence probability to serve as the source or destination of a request than the other nodes in V [40].

In this subsection, we assume that there are two data-center nodes (as shown in Fig. 2.9) in the three EONs for simplicity. Both the two data-center nodes have the same big occurrence probability (45% in this chapter) to be involved in a request (as source or destination), while the other EON nodes equally share the remaining 10% possibility. It should be noted the value of the occurrence probability given here is just for illustrative purpose, which can be in fact arbitrary. We call this distribution the weighted traffic distribution in this chapter.

We denoted by CM_{Ring}^{Wei}, CM_{NSF}^{Wei} and CM_{NJ}^{Wei} the conflict matrices of the Ring, NSFNET and NJ-LATA respectively in the weighted traffic distribution, and the three CMs are as follows.

$$CM_{Ring}^{Wei} = \begin{bmatrix} 0.3829 & 0.1766 \\ 0.1766 & 0.5000 \end{bmatrix}, CM_{NSF}^{Wei} = \begin{bmatrix} 0.3554 & 0.2119 \\ 0.2119 & 0.3982 \end{bmatrix}, CM_{NJ}^{Wei} = \begin{bmatrix} 0.2758 & 0.0616 \\ 0.0616 & 0.3306 \end{bmatrix}.$$

The GOFs of the three EONs and the corresponding minimum intersecting probabilities in the weighted distribution are shown as follows.

- For the Ring

$$p = 0.3829 \times p_1^2 + 0.3532 \times p_1 p_2 + 0.5000 \times p_2^2, \tag{2.14}$$

  the minimum intersecting probability $p_{min} = 30.26\%$ when $p_1 = 0.6105$ and $p_2 = 0.3895$.

- For the NSFNET

$$p = 0.3554 \times p_1^2 + 0.4238 \times p_1 p_2 + 0.3982 \times p_2^2, \tag{2.15}$$

  the minimum intersecting probability $p_{min} = 29.30\%$ when $p_1 = 0.5648$ and $p_2 = 0.4352$.

- For the NJ-LATA

$$p = 0.2758 \times p_1^2 + 0.1232 \times p_1 p_2 + 0.3306 \times p_2^2, \tag{2.16}$$

  the minimum intersecting probability $p_{min} = 18.08\%$ when $p_1 = 0.5568$ and $p_2 = 0.4432$.

Now, we compare the minimum intersecting probability of the three EONs under the two traffic distributions in Table 2.5.

Table 2.5: *Comparison of the Minimum Intersecting Probability*

Traffic \ EON	Ring	NSFNET	NJ-LATA
Uniform Distribution	23.28%	9.79%	8.94%
Weighted Distribution	30.26%	29.30%	18.08%

The minimum intersecting probabilities of the Ring and NSFNET under weighted distribution are the two highest ones, 30.26% and 29.30% respectively, while these of the NJ-LATA and NSFNET under uniform distribution are the two lowest ones, 8.94% and 9.79% respectively.

Thus, if taking their own optimal routing schemes and with a same spectrum assignment method, the final MUFIs of the Ring and NSFNET under the weighted distribution should be the two maximums, and these of NJ-LATA and NSFNET under uniform distribution should be the two minimums.

2.6 Numerical Results

As discussed above, we derived the impact of traffic distribution $\mathcal{D}$ and EON topology G on the lightpath routing by combing Theoretic Chains 1 and 2 in Figs. 2.6 and 2.8 respectively. In this section, we verify the effectiveness of the two theoretic chains by simulations:

- The effectiveness of Theoretic Chain 2 in Fig. 2.8, *i.e.*, the conflict coefficients and the computing method for intersecting probability. In other word, whether the theoretical intersecting probability computed by GOF can fit the realistic one.
- The effectiveness of Theoretic Chain 1 in Fig. 2.6, *i.e.*, the intersecting probability itself. In other words, whether the intersecting probability is positively correlated to the final MUFI.

We conduct simulations on the three EONs under both the uniform and the weighted traffic distributions described before. In our simulations, after the lightpath routing, we utilize a same spectrum assignment algorithm MRSA in [71] to assign FS sets to requests. We consider six scenarios in our simulations, which are labeled in Table 2.6.

Table 2.6: *Six Simulation Scenarios*

EON / Traffic	Ring	NSFNET	NJ-LATA
Uniform Distribution	R-U	NSF-U	NJ-U
Weighted Distribution	R-W	NSF-W	NJ-W

In each scenario, we first pre-compute two candidate paths for each source-destination pair, *i.e.*, the first and second shortest paths as mentioned in Section 2.5. Eleven routing schemes will be conducted and compared in each scenario by increasing p_1 (the percentage of requests routed on the first shortest path) from 0 to 1 with a step of 0.1. Hence, there are 66 cases in total for the six scenarios. Meanwhile, for each case, we compare the realistic intersecting probability p with the one theoretically estimated by GOF through substituting the value of p_1 into the corresponding formulation obtained in Section 2.5. Here the realistic intersecting probability is $\frac{\#\{e\}}{\binom{n}{2}}$, where $\#\{e\}$ is the number of realistic edges in the conflict graph.

Besides, from viewpoint of intersecting probability, we shall also uniformly analyze the six scenarios and 66 cases as follows.

- The final RSA performance of the six scenarios under their own optimal routing schemes.
- The performance differences in the 66 cases.

For the bandwidth range and guard band size, we set $\alpha = 1$, $\beta = 4$, and $GB = 1$ respectively. The number of requests is set as $n = 1000$ in each simulation. We repeat

each simulation 50 times under the same circumstance to ensure sufficient statistical accuracy, and a 95% confidence interval is given to each numerical result. All the simulations have been run by MATLAB 2015a on a computer with 3.2 GHz Intel(R) Core(TM) i5-4690S CPU and 8 GBytes RAM.

2.6.1 Uniform Traffic Distribution

We first verify the three scenarios with the uniform traffic distribution: R-U, NSF-U and NJ-U. The corresponding results are demonstrated in Figs. 2.10(a), 2.10(b) and 2.10(c) respectively.

From these results, we can observe that in all the three scenarios, the realistic intersecting probabilities (marked by blue lines) fit perfectly with the theoretic ones (marked by red lines) that we can barely see the blue lines in the three figures.

The results of intersecting probabilities prove the effectiveness of our Theoretic Chain 2 in Fig. 2.8, *i.e.*, the conflict coefficients and the computing method of GOF.

For R-U and NSF-U, with the intersecting probabilities gradually declining, the corresponding MUFIs decrease and reach the minimum when $p_1 = 1$ in Figs. 2.10(a) and 2.10(b), *i.e.*, their own optimal routing schemes. For NJ-U in Fig 2.10(c), as we can observe that the intersecting probability p is within a narrow range of $[8.94\%, 11.57\%]$ which means the difference of intersecting probabilities is less than 3%. Thus, the difference of the corresponding conflict graphs is so subtle that only a small volatility of MUFIs ([357.38, 379.06]) is observed.

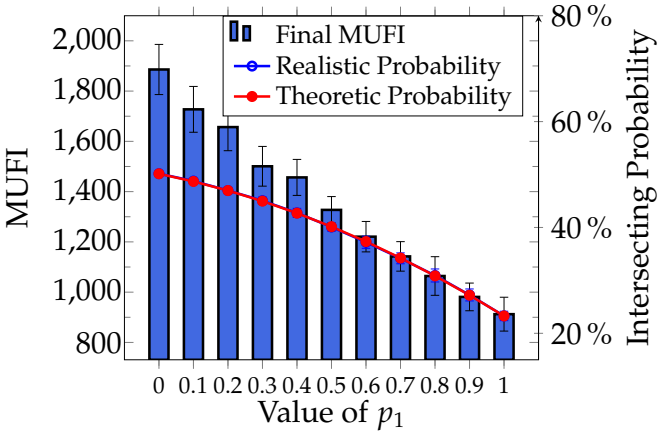
This results of MUFIs mainly prove the effectiveness of our Theoretic Chain 1 in Fig. 2.6, *i.e.*, the intersecting probability is positively correlated to the final MUFI. Further, the results in NJ-U also reflect some insufficiency of the intersecting probability, which needs further investigation on small difference among intersecting probabilities.

2.6.2 Weighted Traffic Distribution

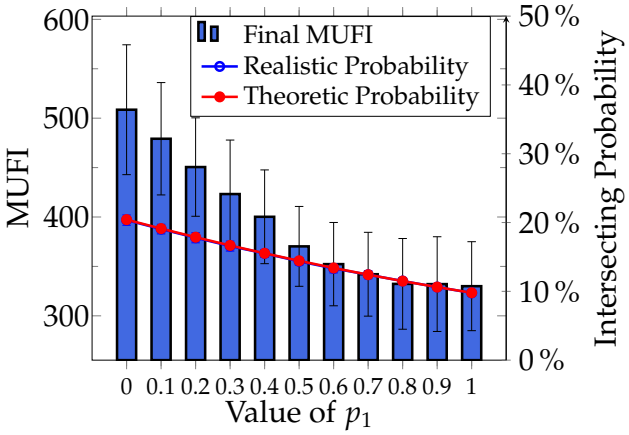
Here, we evaluate the other three scenarios with the weighted traffic distribution: R-W, NSF-W and NJ-W. The corresponding results are demonstrated in Figs. 2.11(a), 2.11(b) and 2.11(c) respectively.

Similarly, the realistic intersecting probabilities match very well with the theoretical one in the three scenarios, which again prove the effectiveness of Theoretic Chain 2 of the conflict coefficients and GOF.

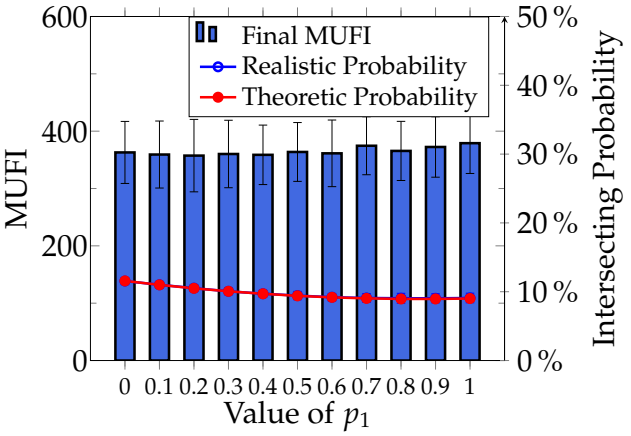
From the aspect of final MUFIs, the three scenarios of R-W, NSF-W and NJ-W represent a common characteristic: First, with the declining of intersecting probabilities, the corresponding MUFIs decrease. After passing some watersheds, the corresponding MUFIs keep increasing as the intersecting probability grows. These results further verify the effectiveness of Theoretic Chain 1 of the intersecting probability.



(a) R-U Scenario.

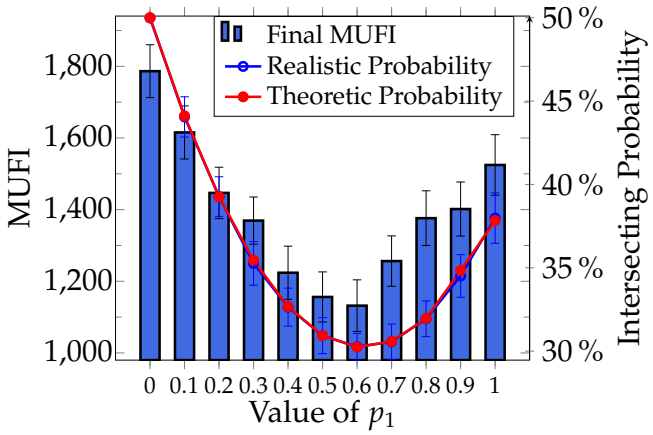


(b) NSF-U Scenario.

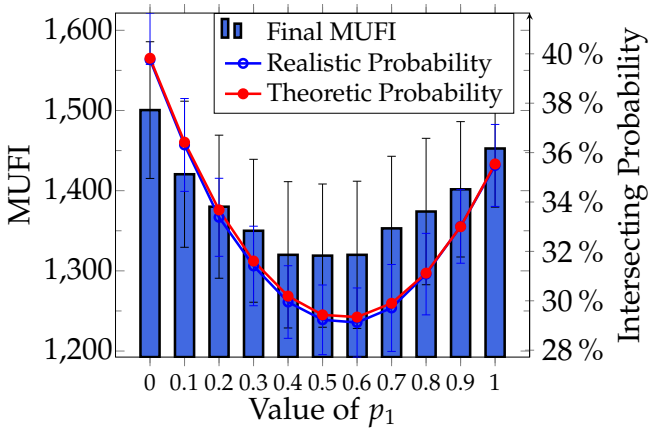


(c) NJ-U Scenario.

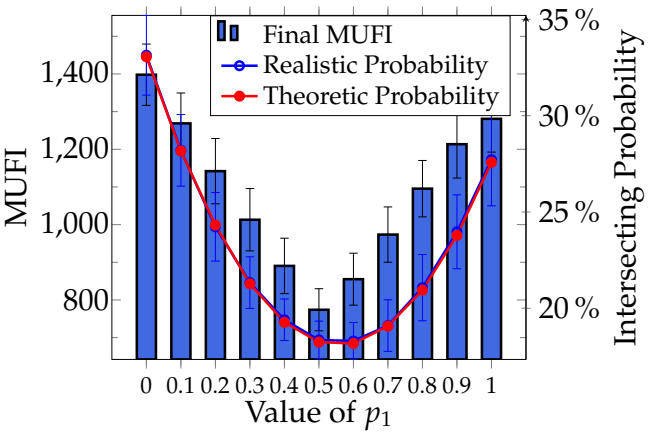
Figure 2.10: *Numerical results for uniform distribution.*



(a) R-W Scenario.



(b) NSF-W Scenario.



(c) NJ-W Scenario.

Figure 2.11: *Numerical results for weighted distribution.*

Meanwhile, the results also exhibited the importance of designing efficient routing scheme to decrease the intersecting probability. When the information of the traffic distribution and network topology are obtained, how to optimally assign the requests on the K candidate paths to reduce the intersecting probability is crucial to the final performance of RSA. Taking the three scenarios of R-W, NSF-W and NJ-W with $K = 2$ for examples, the (worst, best) MUFI pairs are (1786.51, 1132.32), (1500.23, 1319.02) and (1398.41, 774.08) respectively. Thus, reducing the intersecting probability can obtain a huge gain in the final RSA performance.

2.6.3 Comparisons in the Frame of Intersecting Probability

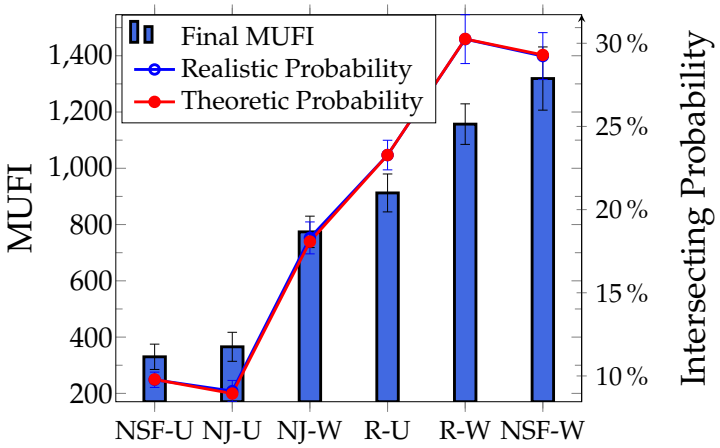
In this subsection, we compare the intersecting probability of the six scenarios using their own optimal routing schemes in Table 2.5. Besides, we present the numerical results of all the 66 cases in the panoramic Fig. 2.12(b) with intersecting probabilities as the X-axis and MUFIs as the Y-axis to show clearly their correlation.

In Fig. 2.12(a), we present the final MUFIs of the six scenarios using their own optimal routing schemes. Compared to the results in Figs. 2.10(a)-2.11(c), it can be observed that the MUFI of the optimal routing scheme of each scenario is indeed the minimum. The realistic intersecting probability, as always, matches very well the theoretic one, which once again proves the effectiveness of the Theoretic Chain 2 of the conflict coefficients and GOF.

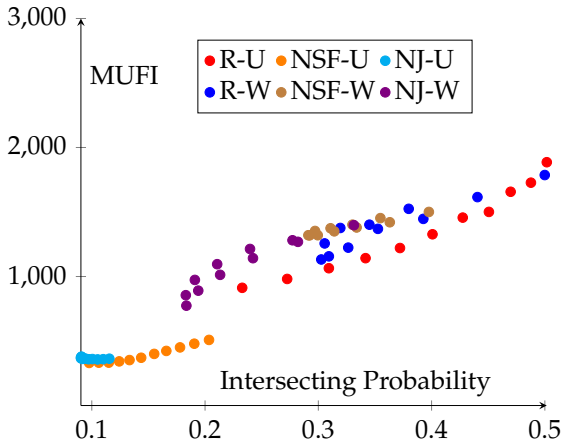
From the final MUFIs of Fig. 2.12(a), we can see that the intersecting probabilities are in general positively correlated to the final MUFIs except two cases: NSF-U vs. NJ-U and NSF-W vs. R-W. This exception can be interpreted as follows. We can see that the intersecting probability difference between NSF-U and NJ-U is less than 1% (similar for NSF-W and R-W). Similar with the phenomena in Fig. 2.10(c), the effectiveness of the intersecting probability will become insignificant when two intersecting probabilities are very close (within 3% gap from our simulations), which needs more delicate works to figure out.

To give a panoramic picture of the relation between intersecting probabilities and final MUFIs, we collect together in Fig. 2.12(b) the numerical results of the 66 cases. Although the 66 cases are conducted in different EONs, traffic distributions as well as routing schemes, we can uniformly analyze their differences in terms of the intersecting probability. Figure 2.12(b) further confirms the Theoretical Chain 1 in Fig. 2.6.

In summary, the numerical results validated the proposed Theoretical Chain 1 in Fig. 2.6 and Theoretical Chain 2 in Fig. 2.8. With the help of the key role, intersecting probability, the two theoretical chains exactly figure out the impact of traffic distribution and EON topology on the lightpath routing. They demonstrate the importance of decreasing the intersecting probability in order to find the optimal routing decision.



(a) Numerical results for the six scenarios using their optimal routing schemes.



(b) The MUFIs and intersecting probabilities in the 66 cases.

Figure 2.12: *Comparisons from intersecting probabilities.*

2.7 Conclusions

In this chapter, we provided a theoretical analysis to reveal the impact of traffic distribution and EON topology on the lightpath routing of RSA problem in EONs. In this sophisticated theoretical analysis, we developed two theoretical chains to figure out the mechanism.

We investigated first the property of the conflict graph built upon the computed lightpaths, since it permits measure to the quality of the RSA. We proved that the optimal MUFI of a conflict graph is directly determined by its chromatic number, and the later one has a strongly positive correlation to the edge existence probability in the conflict graph, *i.e.*, lightpath intersecting probability between any two requests. In other words, the smaller the intersection probability, the smaller optimal MUFI for the RSA, which constitutes our first theoretical chain.

We then characterized the impact of traffic distribution and EON topology by a matrix of conflict coefficients, which together with the routing scheme determines the intersecting probability of any two requests. In order to minimize the intersecting probability so as to minimize the optimal MUFI for RSA, we further developed the quadratic programming GOF to determine the optimal routing scheme. This constitutes our second theoretical chain. Finally, the proposed theoretical chains have been validated by extensive simulations in several well-known EONs.

Chapter 3

Distance Spectrum Assignment in Elastic Optical Networks

Contents

3.1 Introduction

In this chapter, we devote to another important subproblem of the RSA, spectrum assignment. We investigate such spectrum assignment in EONs that the sizes of guard-band frequencies vary.

After all requests in an EON have been routed on their own lightpaths, in order to minimize the potential physical-layer security threat due to inter-channel crosstalk [48], the spectrum assignments of two lightpaths should be separated by a sufficient guard-band when their routing paths share one or more fiber links [82].

For instance, in Fig. 3.1, there are three lightpath requests in an EON, *i.e.*, R_1, R_2 and R_3, and their bandwidth requirements are 2, 4 and 3 FS, respectively. The spectrum assignments of these lightpaths are illustrated at the bottom of Fig. 3.1 with blocks in different colors.

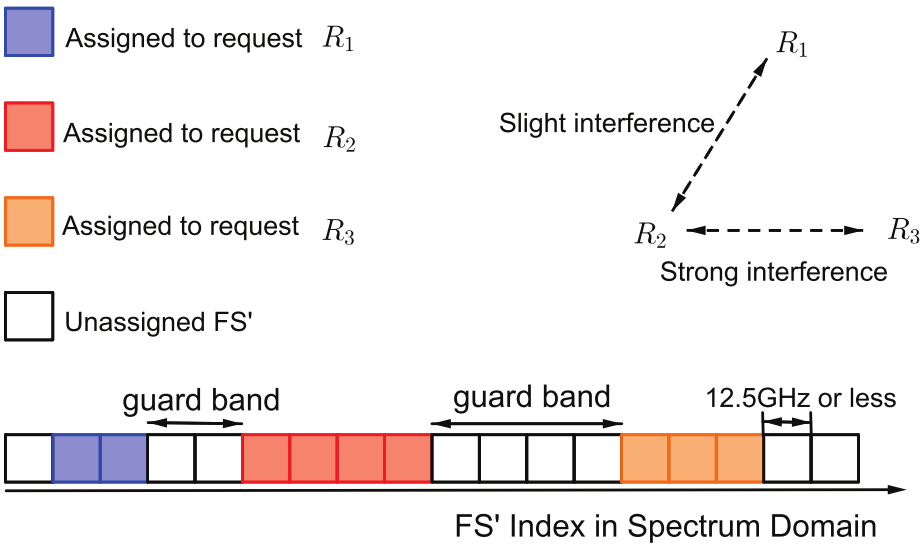


Figure 3.1: *Spectrum assignments with guard-bands in EONs.*

These guard-bands, as shown in Fig. 3.1, can have different sizes, which are not trivial since they determine the impact of inter-channel crosstalk between the lightpaths. In general, the stronger the crosstalk level is or the higher the security requirement is, a larger sized guard-band should be applied. Since the crosstalk level can be affected by many factors such as the required bandwidth, the number of common fiber links and the lightpaths' modulation-levels [44] while the security requirement would depend on the defense of various physical-layer attacks, *e.g.*, eavesdropping and jamming attacks [26], the actual guard-band requirements in EONs would change for different lightpath pairs. Nevertheless, the guard-bands' sizes and the way in which we deploy them would generate spectrum fragmentation and hence significantly influence the spectrum utilization in EONs [60, 80].

Therefore, the service provisioning scheme that uses guard-bands with constant sizes [29, 71] might not be suitable to handle the situation in which the crosstalk levels and/or the security requirements of lightpath pairs are diverse. For instance, a fix-sized guard-band might be insufficient to mitigate a strong crosstalk level while result in spectrum waste for satisfying a relatively low security requirement. Hence, it would be relevant to study how to realize spectrum assignments with various guard-band sizes efficiently.

In this chapter, we put forward a new spectrum assignment model, which uses guard-bands with different sizes to adapt to the crosstalk level or the security requirement of each lightpath pair in an EON. Our model is named as distance spectrum assignment (DSA). We consider the network planning problem in which all the lightpath requests and their routing paths are known, the spectral resources in the EON are sufficient to serve all the requests, and the mutual crosstalk levels or security requirements of the lightpath pairs are also known. With all the aforementioned information, DSA tries to achieve efficient spectrum assignment. Note that, to the best of our knowledge, the problem described by DSA has never been studied theoretically in the literature. Moreover, as we will explain in the chapter, it is an extremely challenging problem. Hence, we explore the characteristics of the DSA problem and provide some interesting and insightful theoretical results to support future studies in this direction. Our theoretical approach mainly revolves around the conflict graph of the DSA, in which each vertex represents a lightpath while an edge signifies the guard-band requirement between two lightpaths. The contributions of this work can be summarized as follows.

- To the best of our knowledge, this is the first work to formally study the DSA problem. We prove the $\mathcal{NP}$-hardness of the problem and analyze its inapproximability, and also formulate an integer linear programming (ILP) model to solve it exactly.
- We formally provide the upper and lower bounds of the optimal solution of DSA and prove that they are tight.
- We propose a two-phase algorithm to solve the DSA problem time-efficiently, and study its performance in various DSA situations, which are represented by different conflict graphs. In its first phase, the algorithm generates an initial solution, which is proven to be the optimal solution in bipartite conflict graphs and can guarantee an approximate ratio of $\mathcal{O}(\log|V|)$ in complete conflict graphs. The second phase improves the initial solution with a random optimization procedure, whose convergence performance are also analyzed mathematically.

The rest of this chapter is organized as follows. Section 3.2 presents our motivation and the related work. In Section 3.3, we model the DSA problem and analyze its hardness. The upper and lower bounds of the optimal solution of DSA are analyzed in Section 3.4. In Section 3.5, we transform DSA into a permutation-based optimization problem (POP), and with this transformation, the two-phase algorithm is developed in Section 3.6. The performance of the two-phase algorithm is theoretically analyzed in Section 3.7, and the numerical results for performance evaluation are presented in Section 3.8. Finally, Section 3.9 summarizes the paper.

3.2 Motivation and Related Work

Since the channel spacing in EONs becomes much narrower than that in WDM networks, the usage of guard-bands, *i.e.*, the unused FS in between the spectrum assignments of two spectrally adjacent lightpaths, becomes more tricky. Specifically, if the guard-bands are not properly chosen, the physical impairments in fibers would induce crosstalk between the lightpaths and thus their Quality of Transmission (QoT) would be deteriorated. Moreover, the crosstalk between two spectrally adjacent lightpaths can be easily utilized to realize physical-layer attacks such as eavesdropping and power jamming [26, 48, 62, 82], and mixed modulation attacks can also degrade the quality of high-bit-rate phase-modulated lightpaths with cross-phase modulation [63]. Therefore, we have to carefully choose the guard-bands to reduce the risk of physical-layer attacks, the degradation of QoT and the nonlinear penalty in EONs [26, 48, 63].

In order to realize spectrally efficient lightpath provisioning, the RSA problem has already been intensively studied. In [71], RSA has been formally defined along with the discussion on its complexity, and an ILP model and two time-efficient heuristics have been designed to solve the RSA problem. The authors of [29] have considered to provision multicast requests in EONs with the multicast-capable Routing, Modulation-level, and Spectrum Assignment (RMSA). However, most of the previous studies on RSA assumed that the guard-bands use a constant size for all the lightpath pairs. Note that, the work in [44] had already revealed that the filtering characteristics of optical components can make the selection of guard-band sizes extremely sophisticated. Therefore, using a fixed guard-band size does not coincide with the practice and thus the problem of DSA, *i.e.*, the spectrum assignment with various guard-band sizes should be investigated in a timely manner.

In general, the Wavelength Assignment (WA) problem in WDM networks (each request in WA problem is assigned with a fixed-wavelength frequency) and the Spectrum Assignment (SA) problem in EONs (each request in SA problem is assigned with a number of FS, the details of distinction between the WA and SA are shown in Table 3.1) can both be studied by leveraging the graph coloring method [13] in conflict graphs that are constructed based on the routing results of lightpaths. Specifically, WA can be solved by finding the chromatic number of the conflict graph [9, 74] while SA can be solved with the interval chromatic number [61, 71]. Nevertheless, DSA differs from the classical graph coloring problem [13] in two aspects: 1) each vertex in the conflict graph, which represents a lightpath, is assigned with a set of contiguous colors (*i.e.*, FS) according to the bandwidth demand rather than only one color; and 2) the distance of the color sets of two adjacent vertices is no longer one but a positive integer, representing the guard-band requirement, which is not identical for all the vertex pairs. More specifically, DSA is similar to the fractional coloring problem [56], with two differences: 1) contiguous colors should be assigned to each vertex in DSA, while this is not the case for fractional coloring; and 2) various distances between adjacent color sets should be kept in DSA while color sets only need to be disjoint in the latter one. For clarity, Table 3.1 provides the comparison of the four coloring related problems that have been discussed above, *i.e.*, the classical coloring, the fractional coloring, the traditional SA, and

Table 3.1: *Comparison of related coloring problems*

	Classical Coloring [13] (*e.g.*, WA [9])	Fractional Coloring [56]	Traditional SA [71]	DSA (this work)
Vertex Color	One color	A set of colors	A set of colors	A set of colors
Color Contiguity	N/A	No need	Required	Required
Color Distance of Adjacent Vertices	Disjoint	Disjoint	Identical positive integer	Various positive integers

the DSA problem. We can see that DSA is apparently a new combinatorial optimization problem, which has not yet been studied before. In the next section, we will formally define the DSA problem.

3.3 Distance Spectrum Assignment (DSA) Problem

In an EON, a set of FS is available in each optical fiber to carry lightpaths. Hence, efficient spectrum assignment algorithms are needed to optimize the spectrum usages of lightpaths under the spectrum contiguous and non-overlapping constraints [83]. Meanwhile, in DSA, to address the crosstalk level and/or security requirement of each lightpath pair, we need to choose a proper guard-band to insert.

3.3.1 Problem Description

For DSA, we consider the network planning problem in which all the lightpath requests and their routing paths are known, the spectral resources in the EON are sufficient to serve all the requests, and the mutual crosstalk levels or security requirements of the lightpath pairs are also known (*i.e.*, the required guard-band sizes are given for all the spectrally adjacent lightpath pairs). Then, DSA tries to achieve efficient spectrum assignment that can not only accommodate all the lightpath requests to satisfy all the constraints, but also minimize the maximum used FS index (MUFI) in the EON.

To solve DSA, we construct a conflict graph based on the known information regarding the lightpaths. Specifically, we first use a vertex to represent each lightpath and assign a weight to it for its bandwidth demand in FS, and then we connect two vertices with an edge if there would be crosstalk between their lightpaths or a guard-band has to be inserted in between the lightpaths' spectrum assignments due to certain customer-specified security reasons. Note that, a weight is also assigned to each edge in the conflict graph to represent the actual required guard-band size. Figs. 3.2 and

3.3 and Table 3.2 show an illustrative example on how to construct the conflict graph. There are four lightpaths with the information in Table 3.2 and their routing paths in a 4-node ring topology is illustrated in Fig. 3.2. Then, we assume that the guard-band requirements for the lightpaths are shown in Fig. 3.3(a), where for simplicity, we use the number of common links in two lightpaths' routing paths as their guard-band requirement. Note that, previous experimental investigation has suggested that the crosstalk level between two spectrally adjacent lightpaths is positively correlated with the number of common links in their routing paths [44]. For instance, since the routing path of R_1, *i.e.*, B-A-D, shares two common links with that of R_4 (C-B-A-D), the required guard-band size between them would be at least 2 FS. In the conflict graph in Fig. 3.3(a), the number inside a cycle is the bandwidth demand in FS while the number on an edge indicates the required guard-band size. Based on the conflict graph in Fig. 3.3(a), we can figure out that optimal solution of DSA is that in Fig. 3.3(b), where the assigned FS to each lightpath are marked with red braces.

Table 3.2: *Information on Lightpaths*

	Bandwidth Demand	**Route**
Request R_1	3 FS	B-A-D
Request R_2	2 FS	C-B-A
Request R_3	3 FS	A-D-C-B
Request R_4	1 FS	C-B-A-D

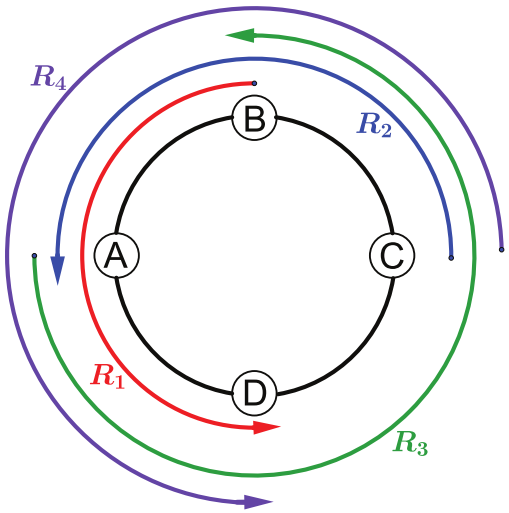
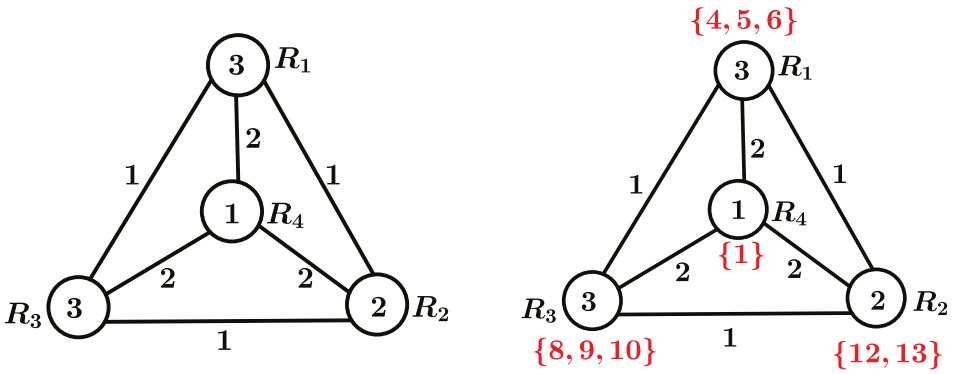


Figure 3.2: *Lightpaths in Table 3.2 in a 4-node ring topology.*

3.3.2 The DSA model and Integer Linear Program

Note that, since we only consider the spectrum assignment problem in DSA, which is already a relatively complex problem as we will explain below, we assume that the rout-



(a) Conflict graph (b) Optimal solution

Figure 3.3: *Conflict graph for lightpaths in Fig. 3.2 and optimal DSA solution.*

ing and guard-band information on the lightpaths are known and thus for each instance of DSA, the conflict graph has already been constructed. Therefore, from now on, we concentrate on how to obtain the optimal spectrum assignments for the lightpaths (*i.e.*, the optimal solution of DSA) based on a known conflict graph, and consider various types of conflict graphs in our analysis. We first introduce the following notations for DSA.

Necessary Notations:

- $G(V, E)$: The DSA conflict graph, where V is the set of vertices, and E is the set of the conflict edges.
- $\mathbb{N}^+$: The set of natural numbers for representing the FS indices in the spectrum domain, which starts from 1.
- v_i: $v_i \in V$ represents the i-th lightpath request.
- v_i^w: The integer weight signifies bandwidth demand of lightpath v_i, in the number of contiguous FS.
- w_{v_i}: The set of contiguous FS assigned to v_i.
- v_i^b: $v_i^b \in \mathbb{N}^+$ is the start-index of w_{v_i}.
- v_i^a: $v_i^a \in \mathbb{N}^+$ is the end-index of w_{v_i}.
- e or v_iv_j: The edge $e \in E$ connecting v_i and v_j, which represents that the lightpaths of v_i and v_j share common link(s). For convenience, we also use v_iv_j to represent an edge e.
- d_e ($d_{v_iv_j}$): The positive integer weight that represents the least guard-band size between lightpaths v_i and v_j.

- B: $B \in \mathbb{N}^+$ is a reasonably large integer.

For ease of discussion, we also use $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$ to represent a DSA graph, *i.e.*, making the weights of vertices and edges explicit. Our objective is to minimize the MUFI in the EON. Note that, it is also possible that the conflict graph G is not a fully connected one. In that case, we can partition G in to a few connected components, solve the DSA problem in them, and then get the MUFI in all the components as the final solution. Hence, we will ignore the cases of non-connected conflict graph in our discussions. The DSA problem can then be defined as

$$\text{Minimize} \quad \max_{s \in \left(\bigcup\limits_{v_i \in V} w_{v_i} \right)} (s) \qquad (\textbf{DSA}), \tag{3.1}$$

where $s \in \mathbb{N}^+$ is the index of a used FS. Meanwhile, DSA should be subject to the following constraints:

- **Bandwidth Requirement Constraint:** Each lightpath should be assigned with enough FS to satisfy its bandwidth demand. In other words, the cardinality of FS set assigned to a vertex $v_i \in V$ should be equal to its weight:

$$|w_{v_i}| = v_i^w, \ \forall v_i \in V, \tag{3.2}$$

- **Spectrum Continuity Constraint:** The FS assigned to a lightpath should be the same on each fiber link in its routing path. Basically, since each lightpath is pre-routed and represented by a vertex in the conflict graph, this constraint will always be satisfied automatically.
- **Spectrum Contiguity Constraint:** The FS assigned to a vertex should be contiguous in $\mathbb{N}^+$, *i.e.*, w_{v_i} can be expressed as $\{v_i^b, v_i^b + 1, ..., v_i^a - 1, v_i^a\}$, where $v_i^b, v_i^a \in \mathbb{N}^+$.
- **Spectrum Set Distance Constraint:** To satisfy the guard-band requirements, the distance between the FS sets assigned to two spectrally adjacent lightpaths should be large enough. Specifically, for each edge $v_i v_j \in E$, the distance between w_{v_i} and w_{v_j} in $\mathbb{N}^+$ should not be smaller than the edge's weight:

$$\text{distance}(w_{v_i}, w_{v_j}) \geq d_{v_i v_j}, \ \forall v_i v_j \in E, \tag{3.3}$$

  where,

$$\text{distance}(w_{v_i}, w_{v_j}) = \min_{s \in w_{v_i}, t \in w_{v_j}} (|s - t| - 1).$$

  One can easy find the difference between the guard band constraint defined in Eq. 2.2 in previous chapter and the distance constraint in Eq. 3.3.

The DSA problem is $\mathcal{NP}$-hard, which will be proven formally in the next subsection. To solve DSA exactly, we formulate an ILP model to obtain the optimal spectrum assignment scheme.

Decision Variables:

- x_i^b: Integer variable to represent the value of v_i^b.
- x_i^a: Integer variable to represent the value of v_i^a.
- y: Integer variable to represent the upper bound of x_i^a.
- $o_{v_iv_j}$: Boolean variable for each edge v_iv_j to represent the order of x_i^b and x_j^b, *i.e.*, if $x_i^b > x_j^b$, we have $o_{v_iv_j} = 1$, and $o_{v_iv_j} = 0$ otherwise.

Objective Function:

$$\begin{aligned} &\text{Minimize} \quad y \quad \text{(ILP-DSA)}, \\ &s.t. \quad \text{Eqs. (3.5)-(3.10)}. \end{aligned} \tag{3.4}$$

$$x_i^a - x_i^b + 1 = v_i^w, \qquad \forall v_i \in V \tag{3.5}$$
$$o_{v_iv_j} + o_{v_jv_i} = 1, \qquad \forall v_iv_j \in E \tag{3.6}$$
$$x_i^a - x_j^b + d_{v_iv_j} + 1 \leq B \times o_{v_iv_j}, \qquad \forall v_iv_j \in E \tag{3.7}$$
$$y \geq x_i^a, \qquad \forall v_i \in V \tag{3.8}$$
$$x_i^a \in \mathbb{N}^+, x_i^b \in \mathbb{N}^+, \qquad \forall v_i \in V \tag{3.9}$$
$$o_{v_iv_j} \in \{0, 1\} \qquad \forall v_iv_j \in E \tag{3.10}$$

3.3.3 Hardness and Inapproximability Analysis

To analyze the hardness of the DSA problem, we introduce the Minimum Hamilton Path problem (MHP) [11], whose objective is to find a minimum Hamilton path in a weighted complete graph. MHP is strongly $\mathcal{NP}$-hard [11].

If the conflict graph of a DSA instance is complete, which means every two vertices v_i and v_j are directly connected. Hence, the FS sets assigned to the lightpaths should be pairwise disjoint. If the complete graph satisfies the triangle inequality, *i.e.*, $d_{v_iv_k} + d_{v_kv_j} \geq d_{v_iv_j}$, $\forall v_i, v_j, v_k \in V$, owing to this inequality, any Hamilton path satisfies the spectrum set distance constraint of DSA. Therefore, in this case, the DSA problem is equivalent to the MHP problem. If the triangle inequality cannot be satisfied in the complete graph, then the solution of DSA might be longer than a Hamilton path. This is because the spectrum set distance constraint might not be satisfied by a Hamilton path. Precisely speaking, the distance between two vertices $v_i, v_j \in V$ in a Hamilton path may be smaller than the required spectrum distance $d_{v_iv_j}$. Theorem 3.3.1 indicates the hardness of DSA.

Theorem 3.3.1. *MHP* $\leq_T^{P1}$ *DSA*

Proof. To prove the $\mathcal{NP}$-hardness of DSA, we just need to prove: 1) any instance $\mathcal{I}$ of MHP can be polynomial-time reduced to an instance $\mathcal{I}'$ of DSA, and 2) the solution of $\mathcal{I}'$ can be converted to that of $\mathcal{I}$ in polynomial time.

[1] $\leq_T^P$ means a polynomial-time Turing reduction.

We get $\mathcal{I}'$ from $\mathcal{I}$ by giving the biggest edge weight b to each vertex of $\mathcal{I}$ as its weight and keeping the edges' weights unchanged. Then, we have $d_{v_iv_k} + b + d_{v_kv_j} \geq d_{v_iv_j}$, $\forall v_i, v_j, v_k$. With this reduction, each vertex pair in a Hamilton path in $\mathcal{I}'$ should satisfy the spectrum set distance constraint. Hence, the solution of $\mathcal{I}$ equals that of $\mathcal{I}'$ minus $|V| \cdot b$, where $|V|$ is the number of vertices. For example, in Fig. 3.4, if we set the weights of the four vertices (*i.e.*, v_1, v_2, v_3 and v_4) to 3, which is the biggest edge weight, the MHP instance $\mathcal{I}$ becomes a DSA instance $\mathcal{I}'$. The minimum Hamilton path can be obtained by solving the DSA instance, which is shown in Fig. 3.4 with red color. The total weight of the minimum Hamilton path is 5, which is obtained by subtracting 12 from the solution of $\mathcal{I}'$. Therefore, we prove that the DSA problem is also $\mathcal{NP}$-hard. □

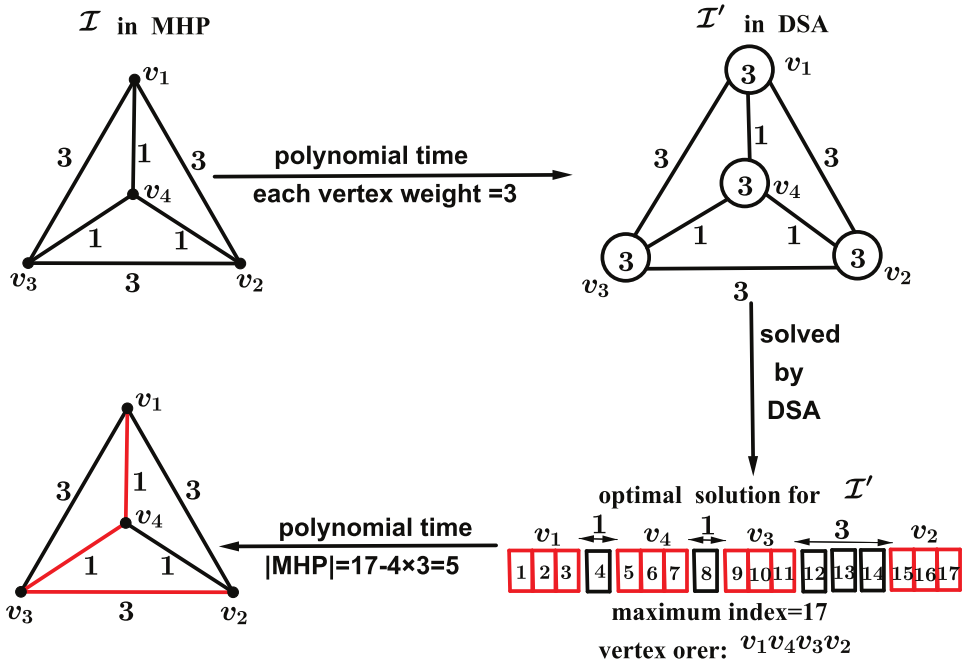


Figure 3.4: *Example on reducing MHP to DSA in polynomial time.*

To analyze the inapproximability of DSA, we first introduce the inapproximability result on the chromatic number $\chi(G)$ of a graph $G(V, E)$. Given a polynomial-time algorithm A to compute the chromatic number of a graph G, we use $A(G)$ to denote the chromatic number obtained by A. The inapproximability of $\chi(G)$ is given by the following statement:

Unless $\mathcal{NP} \subset \mathcal{ZPP}$, no polynomial-time algorithm A that computes the chromatic number of G can guarantee $\frac{A(G)}{\chi(G)}$ within $\mathcal{O}(|V|^{1-\epsilon})$ for an arbitrary graph G, where $|V|$ is the number of vertices in G and $\epsilon > 0$ [24].

Then, we have Theorem 3.3.2.

Theorem 3.3.2. *Unless $\mathcal{NP} \subset \mathcal{ZPP}$, no polynomial-time heuristic algorithm APX for DSA can guarantee $\frac{APX(\mathcal{I})}{OPT(\mathcal{I})}$ within $\mathcal{O}(|V|^{1-\epsilon})$ for all the instances $\mathcal{I}$ of DSA, where $APX(\mathcal{I})$ denotes the MUFI obtained by APX, $OPT(\mathcal{I})$ denotes the optimal result, $|V|$ is the number of vertices in $\mathcal{I}$ and $\epsilon > 0$.*

Proof. For an arbitrary graph $G(V, E)$, we can reduce G to a DSA instance $\mathcal{I}$ in polynomial time by setting the weights of all the vertices and edges as one. Here, we denote this reduction as R, *i.e.*, $R(G) = \mathcal{I}$. According to Theorem 3.4.2, which will be given in the next section, $OPT(R(G)) < 2\chi(G)$. Therefore, if we assume that APX can ensure $\frac{APX(\mathcal{I})}{OPT(\mathcal{I})} < \mathcal{O}(|V|^{1-\epsilon})$ for an arbitrary DSA instance $\mathcal{I}$, $\frac{APX(R(G))}{OPT(R(G))} < \mathcal{O}(|V|^{1-\epsilon})$ would be valid for an arbitrary graph G. Hence, $\frac{APX(R(G))}{2\chi(G)} < \frac{APX(R(G))}{OPT(R(G))} < \mathcal{O}(|V|^{1-\epsilon})$ would be valid for an arbitrary graph G, which means that $APX(R(G))$ can guarantee a ratio within $\mathcal{O}(|V|^{1-\epsilon})$ for $\chi(G)$. This, however, contradicts with the inapproximability of $\chi(G)$. Thus, we get the proof. □

3.4 Upper and Lower Bounds of DSA's Optimal Solution

In this section, we analyze the upper and lower bounds of DSA's optimal solution. For ease of discussion, we first introduce some terminologies and definitions.

- Maximal clique ψ and Maximal clique set Ψ: In a graph $G(V, E)$, a clique $\psi(V_\psi, E_\psi)$ is a maximal clique if and only if there is no clique $\psi' \subseteq G$ and $\psi \subsetneq \psi'$. We use $\Psi(G)$ to denote the set of maximal cliques in G. In the example in Fig. 3.5, there are three maximal cliques ψ_1, ψ_2 and ψ_3, and thus $\Psi(G) = \{\psi_1, \psi_2, \psi_3\}$.
- $C(G)$: The condensation graph of a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$. For the conflict graph, the vertex set V can be partitioned into $\chi(G)$ independent sets. We merge the vertices in the same set as a single super-vertex and assign the maximum weight of the vertices in the set as the weight of the super-vertex. Then, each super-vertex pair in the new graph might have multiple edges. Among these edges, we only keep the one with the biggest weight and remove the others. Finally, we obtain the condensation graph of G.
- $V_{C(G)}$: Set of the vertices in $C(G)$ and $|V_{C(G)}| = \chi(C)$.
- $E_{C(G)}$: Set of the edges in $C(G)$.
- v_i' and $v_i'^w$: $v_i' \in V_{C(G)}$ is the vertex with the i-th biggest weight and $v_i'^w$ is its weight.
- $w_{v_i'}$, $v_i'^b$ and $v_i'^a$: Their definitions are similar as those of w_{v_i}, v_i^b and v_i^a.
- e_i' and $d_{e_i'}$: $e_i' \in E_{C(G)}$ is the edge with the i-th biggest weight and $d_{e_i'}$ is its weight.

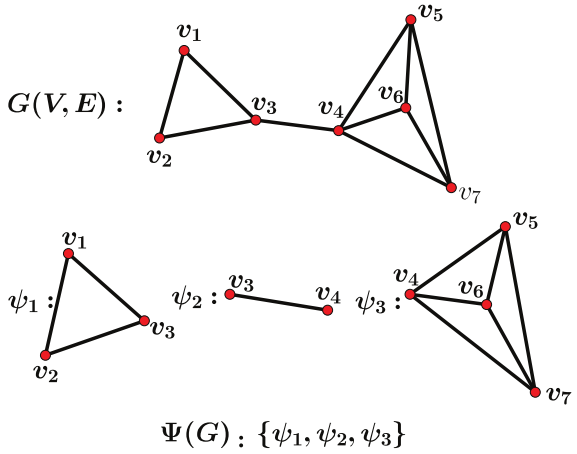


Figure 3.5: *Maximal clique ψ and maximal clique set $\Psi(G)$.*

To study the feature of DSA's optimal solution, we start with the bipartite graphs whose chromatic number $\chi(G)$ is 2. Then, we continue to investigate the connection between the optimal solution and $\chi(G)$ of a DSA conflict graph. Firstly, we give an obvious fact, which is needed in later proofs.

Fact 3.4.1. *Given a DSA conflict graph $G(V, E)$, if there is a solution whose MUFI equals $\max_{v_iv_j \in E}(d_{v_iv_j} + v_i^w + v_j^w)$, then it is an optimal one.*

Proof. Due to the spectrum set distance constraint, the MUFI cannot be less than $d_{v_iv_j} + v_i^w + v_j^w$ for any $v_iv_j \in E$. Hence, if a solution reaches this lower bound, it is optimal. □

We use $opt(G)$ to represent an optimal solution of a DSA conflict graph G, and $|opt(G)|$ to denote the numerical value of the optimal solution, *i.e.*, the optimal MUFI. A *proper spectrum assignment* means that we assign FS sets to the vertices under the four constraints of the DSA problem.

Theorem 3.4.1. *If a DSA conflict graph $G(V, E)$ is a bipartite graph (as shown in Fig. 3.6), where $V = (V_1, V_2)$ with $v_i \in V_1$, $1 \leq i \leq |V_1|$ and $u_j \in V_2$, $1 \leq j \leq |V_2|$, we have $|opt(G)| = \max_{v_iu_j \in E}\{d_{v_iu_j} + v_i^w + u_j^w\}$.*

Proof. Based on Fact 3.4.1, we just need to find a proper spectrum assignment for all the vertices in $G(V, E)$ whose MUFI equals $d_{v_iu_j} + v_i^w + u_j^w$ for certain edge $v_iu_j \in E$. For each $v_i \in V_1$, we assign the FS set w_{v_i} with $v_i^b = 1, v_i^a = v_i^w$. For vertex $u_j \in V_2$, we assign the FS set w_{u_j} with $u_j^b = \max_{v_iu_j \in E}\{v_i^a + d_{v_iu_j} + 1\}$ and $u_j^a = u_j^b + u_j^w - 1$. Because vertices in the same side of the bipartite graph are not adjacent, we can simply verify

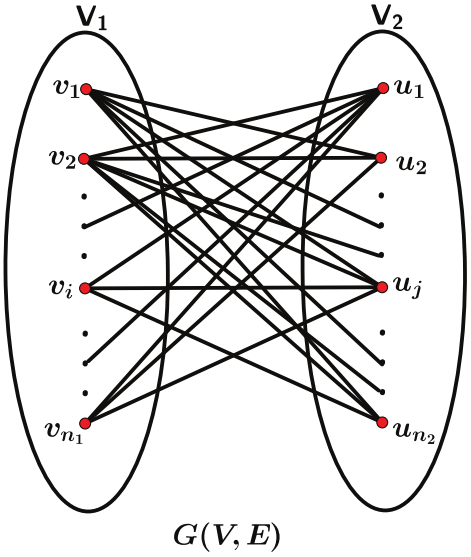


Figure 3.6: *Bipartite DSA graph $G(V, E)$, $V = (V_1, V_2)$.*

that the aforementioned spectrum assignment is proper. Hence, we can see that the final MUFI equals $\max_{v_i u_j \in E} \{d_{v_i u_j} + v_i^w + u_j^w\}$. Hence, we finish the proof. □

Corollary 3.4.1. *If a DSA conflict graph $G(V, E)$ is of such topological structure as tree, even ring or grid, we have $|opt(G)| = \max_{v_i v_j \in E} \{d_{v_i v_j} + v_i^w + v_j^w\}$.*

Proof. Since tree, even ring and grid are bipartite graphs, the proof is trivial based on Theorem 3.4.1. □

Based on the analysis above, we can see that the optimal solution for a DSA conflict graph G with $\chi(G) = 2$ can be obtained easily. Next, the question is how about the conflict graph with $\chi(G) \geq 3$. Apparently, the analysis becomes more difficult for a larger chromatic number. But fortunately, we can get the upper and lower bounds for $|opt(G)|$ by leveraging $\chi(G)$ and the maximal clique ψ. We use MHP(ψ) to represent a minimum Hamilton path in a clique ψ, |MHP(ψ)| to represent its length, and ψ^w to denote the total weight of the vertices in ψ, *i.e.*, $\psi^w = \sum_{v_i \in V_\psi} v_i^w$.

Theorem 3.4.2. *Given an arbitrary conflict graph G, the inequality in Eq. (3.11) holds for the optimal solution of the DSA problem.*

$$\max_{\psi \in \Psi(G)} \{|MHP(\psi)| + \psi^w\} \leq |opt(G)| \leq \sum_{i=1}^{\chi(G)-1} d_{e_i'} + \sum_{i=1}^{\chi(G)} v_i'^w. \tag{3.11}$$

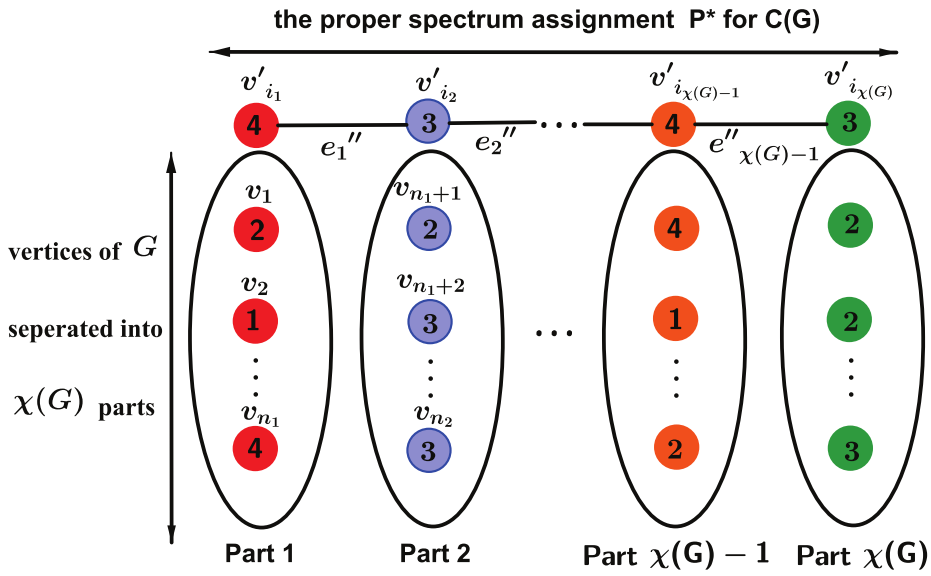


Figure 3.7: *A proper spectrum assignment for $C(G)$.*

Proof. Firstly, we prove $|opt(G)| \leq \sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^{w}_{i}$. To achieve this inequality, we just need to find proper spectrum assignments for all the vertices in $G(V, E)$ and the MUFI would not be bigger than $\sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^{w}_{i}$. Hence, we can first treat $C(G)$ as a conflict graph and find a proper spectrum assignment P^* for $C(G)$. Since each super-vertex $v'_i \in V_{C(G)}$ represents an independent set of G (*i.e.*, its weight is the maximum weight of the vertices in the independent set of G and edge $v'_i v'_j \in E_{C(G)}$ is the largest-weighted edge between the independent sets represented by v'_i and v'_j), we can utilize P^* to find a proper spectrum assignment for G by packing the vertices in v'_i into the FS set $w_{v'_i}$ (as shown in the example in Fig. 3.7). Therefore, if the MUFI of P^* does not exceed $\sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^{w}_{i}$, we prove the inequality. Here, the solution P^* can be obtained with *Algorithm* 3.1. In *Algorithm* 3.1, we start from an arbitrary vertex in $C(G)$, *e.g.*, v'_{i_1}, and get the FS set $w_{v'_{i_1}}$ by setting $v'^{b}_{i_1} = 1$, $v'^{a}_{i_1} = v'^{w}_{i_1}$. We select the largest-weighted incident edge of v'_{i_1}, *e.g.*, $e''_1 = v'_{i_1} v'_{i_2}$, and the corresponding adjacent vertex v'_{i_2} is chosen as the next vertex. Then, we assign $w_{v'_{i_2}}$ by setting $v'^{b}_{i_2} = v'^{a}_{i_1} + d_{e''_1} + 1$, $v'^{a}_{i_2} = v'^{b}_{i_2} + v'^{w}_{i_2} - 1$. After that, we select the largest-weighted incident edge of v'_{i_2} to a vertex whose FS set has not been assigned. The same procedure is repeated until all the vertices in $C(G)$ are assigned with FS sets, and it terminates in $\chi(G) - 1$ steps.

The assignment P^* satisfies all the constraints in DSA, since we select the largest-

Algorithm 3.1: Process Get P^*

Input : $C(G)$
Output: A proper spectrum assignment P^* for $C(G)$

$P^* \leftarrow \emptyset$;
$V'_C \leftarrow$ Random Select v'_{i_1}; % Let v'_{i_1} be Current Vertex
$V'^w_C \leftarrow v'^w_{i_1}$;
$V'^b_C \leftarrow 1$;
$V'^a_C \leftarrow V'^w_C$;
$P^* \leftarrow P^* \cup [V'^b_C, V'^a_C]$;
mark V'_C visited;
while $C(G)$ *still has unvisited vertices* **do**
  search the vertex v' which is the farthest neighbour of V'_C among the unvisited vertices in $C(G)$;
  set e'' as the edge linking v' and V'_C;
  $V'_N \leftarrow v'$; % Let v' be Next Vertex
  $V'^w_N \leftarrow v'^w$;
  $V'^b_N \leftarrow V'^a_C + d_{e''} + 1$;
  $V'^a_N \leftarrow V'^b_C + V'^w_N - 1$;
  $P^* \leftarrow P^* \cup [V'^b_N, V'^a_N]$;
  $V'_C \leftarrow V'_N$;
  mark V'_C visited;
end
return P^*

weighted incident edge in each step. Hence, P^* is a proper spectrum assignment, and the MUFI of P^* equals $\sum_{i=1}^{\chi(G)-1} d_{e''_i} + \sum_{i=1}^{\chi(G)} v'^w_i \leq \sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^w_i$.

Next, we prove the left side. As a maximal clique ψ is a subgraph of G, we have $|opt(\psi)| \leq |opt(G)|$. Hence, we just need to prove $|\text{MHP}(\psi)| + \psi^w \leq |opt(\psi)|$ for any ψ. If we assume that $P(\psi)$ is an optimal proper spectrum assignment for ψ, the FS sets assigned to all the vertices would be mutually disjoint since ψ is a complete subgraph. The distance between any two FS sets in $P(\psi)$ would not be smaller than the weight of the edge connecting the two vertices. Hence, the value of solution $P(\psi)$ would not be smaller than the length of the minimum Hamilton path plus the total weight of all the vertices, *i.e.*, $|\text{MHP}(\psi)| + \psi^w \leq |P(\psi)| = |opt(\psi)|$ and the proof follows. □

In general, it is known that calculating the chromatic number of a graph is extremely difficult. Hence, we provide a more practical method to calculate the bounds. For a graph G, we have $\chi(G) \leq \Delta(G) + 1$ according to the Brook's theorem [13], where $\Delta(G)$ is the maximum degree of G. With a DSA conflict graph $G(V, E, \{v^w_i\}, \{d_{v_i v_j}\})$, we sort the edges and vertices in G in the descending order of their weights, respectively. To avoid confusion, we rename the sorted edges and vertices by denoting the i-th largest-

weighted edge as e_i^s and the vertex with the i-th biggest weight as v_i^s, *i.e.*, $d_{e_i^s} \geq d_{e_j^s}, v_i^{s^w} \geq v_j^{s^w}, \forall i < j, e_i^s, e_j^s \in E, v_i^s, v_j^s \in V$. Then, we have the following corollary.

Corollary 3.4.2. *If* $G(V, E, \{v_i^{s^w}\}, \{d_{e_i^s}\})$ *is a DSA conflict graph, we have* $|opt(G)| \leq \sum_{i=1}^{\Delta(G)} d_{e_i^s} + \sum_{i=1}^{\Delta(G)+1} v_i^{s^w}$.

Proof. Based on the construction procedure of $C(G)$ and the Brook's Theorem, we have $\sum_{i=1}^{\chi(G)-1} d_{e_i'} + \sum_{i=1}^{\chi(G)} v_i'^w \leq \sum_{i=1}^{\chi(G)-1} d_{e_i^s} + \sum_{i=1}^{\chi(G)} v_i^{s^w} \leq \sum_{i=1}^{\Delta(G)} d_{e_i^s} + \sum_{i=1}^{\Delta(G)+1} v_i^{s^w}$. Then, with Theorem 3.4.2, we can verify the proof. □

In order to make a fast estimation for the bounds of the DSA's optimal solution, we can say that the MUFI would not exceed the total weight of $\Delta(G)$ largest-weighted edges plus the total weight of $\Delta(G)+1$ largest-weighted vertices in G.

Corollary 3.4.3. *The two bounds obtained in Theorem 3.4.2 are tight.*

Proof. $\chi(G)$ and ψ are vital for the two bounds. If $G(V, E)$ is a perfect graph [2], then the two bounds can converge under certain conditions. For instance, bipartite graphs are perfect graphs. For a bipartite graph $G(V, E)$, each $\psi(G)$ just contains one edge. As a result, the lower bound $\max_{\psi \in \Psi(G)} \{|\text{MHP}(\psi)| + \psi^w\}$ becomes $\max_{v_i u_j \in E} \{d_{v_i u_j} + v_i^w + u_j^w\}$. In this case, $|opt(G)|$ reaches this lower bound according to Theorem 3.4.1. Moreover, when $\chi(G) = 2$, the upper bound equals $\max_{e \in E}(d_e) + \max_{\forall v_i, \forall u_j} \{v_i^w + u_j^w\}$. When v_i^w and u_j^w are the weights of two adjacent largest-weighted vertices and $d_{v_i u_j}$ is also the maximum weight of the edges, the upper bound equals the lower bound. Hence, the two bounds are tight. □

3.5 Ordered Distance Spectrum Assignment (ODSA)

In order to solve DSA efficiently, we simplify it to an ordered DSA (ODSA) problem, which we will prove that can be solved optimally in polynomial time. Basically, ODSA bears the same objective and constraints of DSA, and besides, it imposes a new **vertex order constraint** as follows. The vertices should be ordered such that the start-FS indices of vertices are in the ascending order, *i.e.*,

$$O_i = (v_{i_1}, v_{i_2}, ..., v_{i_n}) : v_{i_j}^b \geq v_{i_k}^b, \forall j > k \tag{3.12}$$

[2] A graph G is perfect if $\chi(G) = \max_{\psi \in \Psi} |\psi(G)|$ [13].

With the ordered vertices, ODSA becomes a much easier problem than DSA. We formulate the ILP model for ODSA:

$$\begin{aligned} &\text{Minimize } y \quad \text{(ILP-ODSA)}, \\ &s.t. \quad \text{Eqs. (3.14)-(3.18)}. \end{aligned} \tag{3.13}$$

$$x^a_{i_j} - x^b_{i_j} + 1 = v^w_{i_j}, \qquad \forall 1 \le j \le |V| \tag{3.14}$$

$$x^b_{i_j} - x^a_{i_k} \ge d_{v_{i_j} v_{i_k}} + 1, \qquad \forall j > k, v_{i_j} v_{i_k} \in E \tag{3.15}$$

$$x^b_{i_j} \ge x^b_{i_k}, \qquad \forall j > k \tag{3.16}$$

$$y \ge x^a_{i_j}, \qquad \forall 1 \le j \le |V| \tag{3.17}$$

$$x^a_{i_j} \in \mathbb{N}^+, x^b_{i_j} \in \mathbb{N}^+, \qquad \forall 1 \le j \le |V| \tag{3.18}$$

Then, we design a polynomial-time algorithm (O-L) to solve ODSA optimally, and its procedure is shown in *Algorithm* 3.2.

Algorithm 3.2: Procedure of O-L

Input : A DSA graph $G(V, E, \{v^w_i\}, \{d_{v_i v_j}\})$, and a vertex order $O_i = (v_{i_1}, v_{i_2}, ..., v_{i_n})$

Output: An assignment strategy for the star-index Seq $= \{v^b_{i_j} : 1 \le j \le |V|\}$ and the MUFI

$v^b_{i_1} \leftarrow 1$;
$v^a_{i_1} \leftarrow v^w_{i_1}$;
Seq $\leftarrow v^b_{i_1}$;
$j \leftarrow 2$;
while $j \le |V|$ **do**
  $s_1 \leftarrow \max_{\forall k<j, v_{i_k} v_{i_j} \in E} \{v^a_{i_k} + d_{v_{i_k} v_{i_j}} + 1\}$;
  $s_2 \leftarrow v^b_{i_{j-1}}$;
  $v^b_{i_j} \leftarrow \max\{s_1, s_2\}$;
  $v^a_{i_j} \leftarrow v^b_{i_j} + v^w_{i_j} - 1$;
  Seq $\leftarrow$ Seq $\cup \{v^b_{i_j}\}$;
  $j \leftarrow j + 1$;
end
return Seq and $\max_{1 \le j \le |V|} (v^a_{i_j})$

The main idea of *Algorithm* 3.2 is to assign the FS sets to vertices in sequence according to the pre-defined order such that $v^b_{i_j}$ takes the smallest possible value to satisfy all the constraints of ODSA. We begin with vertex v_{i_1}, and set its start-FS index $v^b_{i_1}$ as 1 and its end-FS index according to its bandwidth demand, *i.e.*, $v^a_{i_1} = v^w_{i_1}$. Then, $v^b_{i_1}$ is added to the spectrum assignment scheme Seq. For each vertex v_{i_j}, we use s_1 to

denote the smallest index permitting to keep enough guard-bands from the adjacent vertices of v_{i_j} that have already been assigned FS sets, and use $s_2 = v^b_{i_{j-1}}$ to satisfy the order constraint. Then, the start-index of v_{i_j} is $\max\{s_1, s_2\}$, and the end-index equals $\max\{s_1, s_2\} + v^w_{i_j} - 1$. The procedure terminates when all the vertices have been assigned FS sets. The time complexity of *Algorithm* 3.2 is $\mathcal{O}(|E|)$.

Theorem 3.5.1. Algorithm 3.2 *obtains an optimal FS assignment scheme for ODSA.*

Proof. Firstly, it is easy to verify that the spectrum assignment scheme in Seq is a feasible solution for ODSA, since all the constraints are satisfied. Then, we prove that Seq indicates an optimal ODSA solution by contradiction. If Seq is not optimal, $opt = \{v^{b^{opt}}_{i_j} : 1 \leq j \leq |V|\}$ would be the optimal start-index arrangement for ODSA and $opt \neq$ Seq. Let j be the index of the first vertex such that $v^{b^{opt}}_{i_j} \neq v^b_{i_j}$ ($\because v^{b^{opt}}_{i_1} = v^b_{i_1} = 1 \Rightarrow j \geq 2$). Then, with the greedy strategy, we have $v^{b^{opt}}_{i_j} > v^b_{i_j}$. For $j+1$, let $\mathcal{F}^{opt}_{j+1}$ and $\mathcal{F}_{j+1}$ be the feasible region for $v^{b^{opt}}_{i_{j+1}}$ and $v^b_{i_{j+1}}$ in ILP-ODSA (*cf.*, Eq. (3.13)), respectively. The other constraints are the same for $v^{b^{opt}}_{i_{j+1}}$ and $v^b_{i_{j+1}}$ except for Eq. (3.16), *i.e.*, $v^{b^{opt}}_{i_{j+1}} \geq v^{b^{opt}}_{i_j}$ and $v^b_{i_{j+1}} \geq v^b_{i_j}$. As $v^{b^{opt}}_{i_j} > v^b_{i_j}$, the lower bound of $\mathcal{F}_{j+1}$ denoted as ζ_{j+1} would not be larger than that of $\mathcal{F}^{opt}_{j+1}$ denoted as ζ^{opt}_{j+1}. Since we can get $v^b_{i_{j+1}} = \zeta_{j+1}$ with the greedy strategy of *Algorithm* 3.2, we have $v^{b^{opt}}_{i_{j+1}} \geq \zeta^{opt}_{j+1} \geq \zeta_{j+1} = v^b_{i_{j+1}}$. By induction, we have $v^{b^{opt}}_{i_k} \geq v^b_{i_k}$, where k is within $[j, |V|]$. Therefore, we prove that the MUFI of ODSA under opt arrangement would not be smaller than that is provided by Seq, which causes the contradiction. Then, we finish the proof. □

Corollary 3.5.1. *A DSA problem can be solved optimally with* Algorithm 3.2 *under certain vertex order.*

Proof. If opt is an optimal solution for a DSA problem, there has to be an order O_{opt} among the start-FS indices of opt. Therefore, the optimal solution for ODSA with vertex order O_{opt} equals that of the DSA problem. With Theorem 3.5.1, we prove that *Algorithm* 3.2 can get the optimal solution for the DSA problem under the order O_{opt}. □

Now, we can see that it would be vital to determine the optimal vertex order. Note that, in the analysis above, we actually have already transformed the DSA problem into the Permutation-based Optimization Problem (POP). POP is a classical combinational optimization [68]: Let S be a set of n elements, Σ be the permutation space that consists of $n!$ permutations over S, and $f(\cdot)$ be an estimation function for any $\sigma \in \Sigma$. The objective of POP is to optimize $f(\cdot)$ over Σ.

$$\sigma^* = arg \min_{\sigma \in \Sigma} f(\sigma). \tag{3.19}$$

For the DSA problem, S is vertex set V, Σ is the whole $|V|!$ vertex orders and we can utilize *Algorithm* 3.2 as our estimation function. In the next section, we will get an initial vertex order with a heuristic algorithm and then improve the vertex order with the Nested Partitions Method (NPM) [58].

3.6 Time-Efficient Approximation Algorithm for DSA

For any DSA problem, if the vertex order (*i.e.*, in the ascending order of the start-FS index) in the optimal solution is known beforehand, then it can be transformed into an ODSA problem and solved optimally by *Algorithm* 3.2 in polynomial time. Inspired by this, we develop a two-phase algorithm to solve DSA. Specifically, in the first phase, we use a greedy strategy to generate an initial vertex order, and then the second phase utilizes NPM to improve the initial order.

3.6.1 First Phase Greedy Algorithm (FPGA)

For a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$, we can get the initial vertex order with the following procedure. Firstly, we start from any vertex $v_i \in V$, and find the FS set for v_i with a greedy strategy, *i.e.*, $v_i^b = 1$ and $v_i^a = v_i^w$. Meanwhile, we set a variable O_i to record the order of vertices according to the assigned FS sets. Hence, O_i takes v_i as the first element. Then, we find the vertex v_j from the vertices that are not yet in O_i to ensure that v_j^b is the minimum to satisfy the constraints of DSA for all the vertices that are in O_i. We insert this v_j into O_i and assign the corresponding FS set to it. The same procedure is repeated until all the vertices have been included into O_i. After $|V|$ while-loops, $|V|$ vertex orders $\{O_1, O_2, ..., O_{|V|}\}$ have been generated and we choose the one that results in the minimum MUFI as our initial order. *Algorithm* 3.3 gives the procedure of the proposed First Phase Greedy Algorithm (FPGA). In *Lines* 1-3, starting from $j = 1$, we initialize O_j as $\emptyset$ and use s_j to record the current MUFI used in O_j, whose initial value is 0. Then, in *Lines* 4-20, with the $|V|$ while loops, we generate $|V|$ vertex orders. As mentioned above, *Lines* 5-8 let v_j be added into O_j, assign the FS set to it, and update s_j as $s_j = v_j^w$. In the for-loop covering *Lines* 9-20, we organize the remaining vertices for O_j one by one using the aforementioned greedy strategy. Finally, we select the vertex order that results in the minimum MUFI. We can see that there are three cascading loops in *Algorithm* 3.3, and thus its time complex is $\mathcal{O}(|V|^3 \cdot \Delta)$, where $|V|$ is the number of vertices and Δ is the maximum degree of G.

After getting the initial vertex order O^* and initial MUFI value s^*, we utilize NPM to improve the initial solution. In the next subsection, we will provide the details of NPM and our two-phase algorithm.

3.6.2 Two-phase Algorithm

The NPM method was proposed in [58] to leverage a general random method to solve global optimization problems, which includes POP. Specifically, we consider the following problem

$$\theta^* = arg \min_{\theta \in \Theta} f(\theta), \tag{3.20}$$

where Θ is the entire solution space and $f(\cdot) : \Theta \to \mathbb{R}$ is the objective function. Firstly, NPM gives a *partitioning scheme* to partition Θ systematically, and then it uses a iterative

Algorithm 3.3: Procedure of FPGA

Input : $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$
Output: An initial vertex order and an initial MUFI

$j \leftarrow 1$;
$O_j \leftarrow \emptyset$; % initialize vertex order O_1
$s_j \leftarrow 0$; % record the MUFI of O_1
while $j \leq |V|$ **do**
  $O_j \leftarrow O_j \cup \{v_j\}$;
  $v_j^b \leftarrow 1$;
  $v_j^a \leftarrow v_j^w$;
  $s_j \leftarrow v_j^w$;
  for $i = 2 : |V|$ **do**
    $v \leftarrow \emptyset$; % v is the next vertex entering O_j
    $v^b \leftarrow B$; % B is large enough
    $v^w \leftarrow 0$;
    for $k = 1 : |V|$ **do**
      if $v_k \notin O_j$ **then**
        $v_k^b \leftarrow \max_{\forall v_l \in O_j, v_k v_l \in E} \{v_l^a + d_{v_k v_l} + 1\}$;
        if $v_k^b < v^b$ **then**
          $v \leftarrow v_k$; $v^b \leftarrow v_k^b$; $v^w \leftarrow v_k^w$;
        end
      end
    end
    $O_j \leftarrow O_j \cup \{v\}$;
    $v^a \leftarrow v^b + v^w - 1$;
    $s_j \leftarrow \max\{s_j, v^a\}$;
  end
  $j \leftarrow j + 1$; $O_j \leftarrow \emptyset$; $s_j \leftarrow 0$;
end
$O^* = \underset{1 \leq j \leq |V|}{argmin}^{O_j} s_j$; $s^* = \underset{1 \leq j \leq |V|}{argmin}^{s_j} s_j$;

approach to optimize $f(\cdot)$. In each iteration, NPM operates on a solution space η, which is a subset of Θ from the partitioning scheme and is named as *the most promising* region. Then, according to the partitioning scheme, we divide the most promising region η into $M(\eta)$ disjoint subregions, and we call $\Theta \backslash \eta$ *surrounding region*. Note that, if the partition scheme obtains a region, then we say the region is *valid*, and if a valid region σ is formed by partitioning a valid region η, then σ is a *subregion* of η and η is called the *superregion* of σ. Therefore, η is divided into $M(\eta)$ disjoint subregions. Next, each of the $M(\eta)$ subregions and the surrounding region are sampled by a *random sampling scheme* and we use the objective function to evaluate the samples and calculate the *promising index* for

each subregion. If the promising index of a subregion among the $M(\eta)$ subregions of η turns to be the best one, we set this subregion as the most promising region in the next iteration. If the surrounding region is proven to be the best, the method will *backtrack* to another region to be the next most promising region (*e.g.*, a region that contains the previous most promising region or a subregion of Θ that contains the best sample). The most promising region is then partitioned and sampled with the procedure discussed above.

For DSA, the entire solution space Θ is the $|V|!$ vertex orders and the objective function is *Algorithm* 3.2. In the second phase of the time-efficient approximation algorithm for DSA, the partitioning scheme is as follows: we first divide Θ into n disjoint subregions by choosing $v_1, v_2, ..., v_{|V|}$ as the first vertex in the ordered vertices, and then each of the $|V|$ subregions is divided into $|V| - 1$ subregions by selecting the second vertex and so on so forth. Fig. 3.8 provides an illustrative example on the partitioning scheme for DSA. The random sampling scheme samples the surrounding region and each subregion uniformly and the most promising region will backtrack to the least supperregion if the promising index in the surrounding region is the best. The vertex order O^* obtained by *Algorithm* 3.3 is the original most promising region.

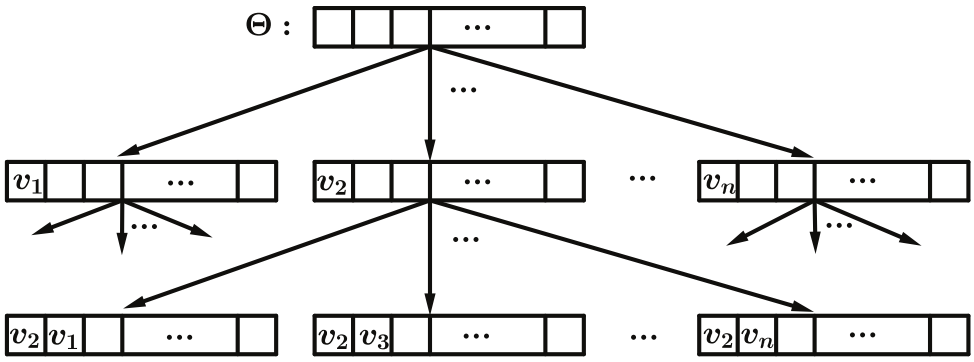


Figure 3.8: *Example on the partitioning scheme for DSA.*

3.7 Algorithm Analysis

In this section, we analyze the performance of the two-phase algorithm, which is composited by *Algorithm* 3.3 (*i.e.*, an approximation algorithm) and NPM (*i.e.*, a random optimization algorithm). For *Algorithm* 3.3, as DSA is intractable according to Theorem 3.3.2, we focus the analysis on some specific graph types, *e.g.*, complete graphs and bipartite graphs. For NPM, we provide two of its key properties, *i.e.*, the convergence performance and the number of expected iterations.

3.7.1 Approximate Ratio of FPGA in Special Graphs

Complete Graph with Triangle Inequality

If a DSA conflict graph $G(V, E)$ is a complete graph, the FS sets assigned to the vertices must be pairwise disjoint. Hence, to satisfy the bandwidth requirement and spectrum contiguity constraints, the union of the FS sets assigned to the vertices has a fixed cardinality, denoted as $V^w = \sum_{i=1}^{n} v_i^w$. Consequently, the optimization objective in this case is equivalent to minimize the sizes of the guard-band between any two spectrally adjacent FS sets under the spectrum set distance constraint.

An algorithm, called *Nearest Neighbor* (NN) to solve MHP, can guarantee an approximate ratio for complete conflict graphs that satisfy the triangle inequality. *Algorithm* 3.4 shows the procedure of NN.

Algorithm 3.4: Procedure of NN

Input : $G(V, E), v_i, \{d_{v_i v_j}\}$
Output: A Hamilton path P

```
1 set CurrentVertex ← v_i;
2 mark v_i visited;
3 while G still has unvisited vertices do
4 |   find vertex v which is the nearest neighbor to CurrentVertex among all the
  |     unvisited vertices in G;
5 |   CurrentVertex ← v;
6 |   mark CurrentVertex visited;
7 |   P ← P ∪ {v};
8 end
9 return P
```

Let $|NN(G)|$ denote the length of the Hamilton path produced by *Algorithm* 3.4 and $|MHP(G)|$ denote the length of the minimum Hamilton path. Then according to [11, 39], we have the approximate ratio $\frac{|NN(G)|}{|MHP(G)|} \leq \frac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. For a complete DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$ that satisfies the triangle inequality, we apply *Algorithm* 3.3 to G. Note that, the while-loop in *Algorithm* 3.3 obtains a vertex order O_j in the j-th iteration. There is a proper spectrum assignment induced by O_j, which in fact represents a Hamilton path in G. Then, we have Lemma 3.7.1.

Lemma 3.7.1. *If the conflict graph G is a complete graph that satisfies the triangular inequality, the Hamilton path induced by the order O_j from* Algorithm *3.3 is equivalent to the result from* Algorithm *3.4 with input v_j.*

Proof. We assume that the order O_j obtained in the j-th while-loop of *Algorithm* 3.3 is $(v_{j_1}, v_{j_2}, ..., v_{j_n})$, where $v_{j_1} = v_j$. At first we have $|O_j| = 1$, which means that only v_{j_1} is included in order O_j. Then, with the greedy strategy of *Algorithm* 3.3, v_{j_2} is the nearest

neighbor to v_{j_1} in G. Supposing this inference is true when $|O_j| = k$, where $k < |V|$, we assert $v_{j_{k+1}}$ is the nearest neighbor of v_{j_k} among those vertices that are not yet in O_j. After we have included the first k vertices in O_j, the innermost for-loop of *Algorithm* 3.3 searches the $(k+1)$-th vertex *i.e.*, $v_{j_{k+1}}$, whose FS start-index is the smallest among those unordered vertices. We use v_l to denote the nearest neighbor of v_{j_k} among all the unordered vertices. Since the triangle inequality is held, the spectrum set distance constraint for $v_{j_{k+1}}$ only comes from v_{j_k}, *i.e.*, $v^b_{j_{k+1}} = v^a_{j_k} + d_{v_{j_k}v_{j_{k+1}}} + 1$. As $d_{v_{j_k}v_l}$ is the smallest guard-band size, $v^b_{j_{k+1}} = v^a_{j_k} + d_{v_{j_k}v_l} + 1$ reaches the minimum. Therefore, using the greedy strategy, we can get $v_{j_{k+1}} = v_l$ and the proof is verified. □

In fact, we select the minimum one from $\{O_1, O_2, ..., O_n\}$ after $|V|$ while-loops in *Algorithm* 3.3. Let $|FPGA(G)|$ be the finial output value and $|opt(G)|$ be the optimal value for a DSA conflict graph G. Then, according to Lemma 3.7.1 and the analysis above, $|FPGA(G)| - V^w$ and $|opt(G)| - V^w$ are the length of the Hamilton path produced by *Algorithm* 3.4 and $|MHP(G)|$ respectively. Then, we get the following theorem.

Theorem 3.7.1. *If $G(V, E, \{v^w_i\}, \{d_{v_iv_j}\})$ is a complete DSA conflict graph that satisfies the triangle inequality, the approximate ratio of* Algorithm 3.3 *would not be larger than* $\frac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$.[3]

Proof. According to the analysis above, we have $\frac{|FPGA(G)| - V^w}{|opt(G)| - V^w} \leq \frac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. As $|FPGA(G)| \geq |opt(G)|$, we have $\frac{|FPGA(G)|}{|opt(G)|} \leq \frac{|FPGA(G)| - V^w}{|opt(G)| - V^w} \leq \frac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. □

Bipartite Graphs

Then, we consider the case in which the DSA conflict graph is a bipartite graph. Before the analysis, we introduce the following definition.

Definition 3.7.1. *For a bipartite graph $G(V_1, V_2)$, V_1 and V_2 are the two parts of the vertices in G. We call* ***its vertex labeling is good*** *if the vertices are labeled in the way that the vertices in V_1 are labeled as the first $|V_1|$ ones,* i.e., *$\{v_1, v_2, ..., v_{|V_1|}\} = V_1$, and apparently, the remaining vertices are all in V_2 and labeled as $\{v_{|V_1|+1}, v_{|V_1|+2}, ..., v_{|V|}\} = V_2$. For a bipartite graph $G(V_1, V_2)$, the time needed to get a good vertex labeling is $\mathcal{O}(|E|)$.*

Theorem 3.7.2. *If a DSA conflict graph $G(V, E, \{v^w_i\}, \{d_{v_iv_j}\})$ is a bipartite graph and we label its vertices in a good way,* Algorithm 3.3 *can get the optimal solution for DSA.*

Proof. Let V_1 and V_2 be the two parts of a bipartite V. According to *Algorithm* 3.3 and Theorem 3.4.1, we just need to prove the MUFI obtained with order O_1 in *Algorithm* 3.3 equals $\max_{v_iv_j \in E}\{d_{v_iv_j} + v^w_i + v^w_j\}$. After v_1 has entered O_1, since V_1 is an independent set, *Algorithm* 3.3 includes vertices $v_2, ..., v_{|V_1|}$ in O_1 in sequence and $v^b_i = 1, 1 \leq i \leq |V_1|$.

[3] Actually, for this special case, *Double Minimum Spanning Tree* algorithm [54] of MHP can be utilized for DSA, which can guarantee a 2-approximation ratio with the similar proof of Theorem 3.7.1.

Also, because V_2 is an independent set, $v_i^b = \max_{\forall v_j \in V_1, v_i v_j \in E} \{v_j^a + d_{v_i v_j} + 1\}, |V_1| + 1 \leq i \leq |V|$. Therefore, considering the four constraints of DSA, we get the MUFI of O_1 as $\max_{v_i v_j \in E} \{d_{v_i v_j} + v_i^w + v_j^w\}$. □

3.7.2 Convergence Performance and Expected Number of Iterations of Two-phase Algorithm

Convergence Performance

For *Algorithm* 3.3, we assume that the partitioning scheme has been defined and let Σ denote the set of all the valid regions, where $\sigma(0)$ is the initial region state, *i.e.*, the initial vertex order that is obtained, and $\sigma(k) \in \Sigma$ is the region state of the k-th iteration. Then, $\{\sigma(k)\}_{k=0}^{\infty}$ is the iteration sequence and the region state $\sigma(k+1)$ depends on the estimated values of the promising index in the state $\sigma(k)$, which is related with the sampling points. Therefore, $\{\sigma(k)\}_{k=0}^{\infty}$ is a Markov chain with state space Σ, and we have Theorem 3.7.3 according to [58].

Theorem 3.7.3. *$\eta \in \Sigma$ is an absorbing state of the Markov chain $\{\sigma(k)\}_{k=0}^{\infty}$, **if and only if** η is the optimal vertex order for DSA.*

Proof. Firstly, we prove the "if" part and use Algorithm 3.2 as the object function $f(\cdot)$ to evaluate the promising index of a region. If we assume that η is the optimal vertex order for DSA, then the transition probability of staying in η is: $P_{\eta\eta} = P[f(\eta) \leq f(\Theta \backslash \eta)] = 1$. Hence, η is an absorbing state. Next, we prove the reverse. Supposing ξ is an absorbing state and ξ does not represent the optimal order for DSA, the transition probability of not staying in ξ is: $P_{\xi\Theta\backslash\xi} = P[f(\xi) > f(\Theta\backslash\xi)] \geq P$[randomly select a point θ in $\Theta\backslash\xi$ and $f(\theta) < f(\xi)] > 0$. This inequality reveals that ξ is a transient state, which leads to a contradiction. Therefore, we finish the proof. □

According to Theorem 3.7.3, the Markov chain will eventually converge to an optimal vertex order and stay there forever. Since the transient states are finite, we can see that the Markov chain would reach an optimal vertex order within finite time.

Expected Number of Iterations

The expected number of iterations to reach the optimal vertex order directly impacts the time-efficiency of our two-phase algorithm. To evaluate the expected number of iterations, we need to introduce several random variables and symbols [59]. We use Σ to represent the state space, σ_{opt} to represent the optimal solution regions, *i.e.*, the optimal vertex order. We define $\Sigma_1 = \{\eta \in \Sigma \backslash \{\sigma_{opt}\} | \sigma_{opt} \in \eta\}$, *i.e.*, the valid regions that include σ_{opt} and $\Sigma_2 = \{\eta \in \Sigma \backslash \{\sigma_{opt}\} | \sigma_{opt} \notin \eta\}$, *i.e.*, the valid regions that do not include σ_{opt}. Then, we have $\Sigma = \{\sigma_{opt}\} \cup \Sigma_1 \cup \Sigma_2$. We use Y_η to denote the number of visits of a state $\eta \in \Sigma$ and use T_η to represent its hitting time (the first time of visiting

this state). Besides, we denote the probability of an event under constraint that the chain starts in a state $\eta \in \Sigma$ as $P_\eta[\text{event}]$.

According to [59], the number of iterations for the Markov chain to reach an absorbing state Y equals the number of iterations to visit all the transient states plus one (*i.e.*, the transition to the absorbing state), which is $Y = 1 + \sum_{\eta \in \Sigma_1} Y_\eta + \sum_{\eta \in \Sigma_2} Y_\eta$. As Σ is finite, we get the expected number of iterations as

$$\mathbf{E}[Y] = 1 + \sum_{\eta \in \Sigma_1} \mathbf{E}[Y_\eta] + \sum_{\eta \in \Sigma_2} \mathbf{E}[Y_\eta]. \tag{3.21}$$

Theorem 3.7.4. *Let $\sigma(0)$ be the initial vertex order provided by* Algorithm 3.3. *The expected number of iterations for our two-phase algorithm to get the optimal solution for DSA is*

$$\boldsymbol{E}[Y] = 1 + \sum_{\eta \in \Sigma_1} \frac{1}{P_\eta[T_{\sigma_{opt}} < T_\eta]} + \sum_{\eta \in \Sigma_1} \frac{P_{\sigma(0)}[T_\eta < min\{T_{\sigma(0)}, T_{\sigma_{opt}}\}]}{P_\eta[T_{\sigma(0)} < T_\eta] \cdot P_{\sigma(0)}[T_{\sigma_{opt}} < min\{T_{\sigma(0)}, T_\eta\}]}. \tag{3.22}$$

Proof. As given in [58], the expected number of visits to the transient states is

$$\mathbf{E}[Y_\eta] = \begin{cases} \dfrac{1}{P_\eta[T_{\sigma_{opt}} < T_\eta]}, \eta \in \Sigma_1, \\ \dfrac{P_{\sigma(0)}[T_\eta < \min\{T_{\sigma(0)}, T_{\sigma_{opt}}\}]}{P_\eta[T_{\sigma(0)} < T_\eta] \cdot P_{\sigma(0)}[T_{\sigma_{opt}} < \min\{T_{\sigma(0)}, T_\eta\}]}, \eta \in \Sigma_2. \end{cases} \tag{3.23}$$

By substituting Eq. (3.23) in Eq. (3.21), we finish the proof. □

In each iteration, we at most take n sampling points in the n valid regions. Each sampling and calculating of the promising index will use the Procedure O-L whose time complexity is $\mathcal{O}(|E|)$. Therefore, the expected time complexity for the second phase is $\mathcal{O}(|V| \cdot |E| \cdot \mathbf{E}(Y))$.

Although we have Theorem 3.7.4, calculating the expected number is still tough. Hence, we leverage the approximation stochastic model in [59]. Specifically, in each iteration, if the promising index of the surrounding region is the best, we backtrack to the entire solution space Θ. Let P_0 be the the probability of the two-phase algorithm moving towards the correct direction, *i.e.*, backtracking if the optimal solution is not in the current most promising region and selecting the correct subregion otherwise. Then, we have Theorem 3.7.5.

Theorem 3.7.5. *Assuming the above approximation stochastic model is held, the expected number of iterations for two-phase algorithm to find the optimal solution for DSA is*

$$\boldsymbol{E}(|Y|) = \frac{1}{P_0^n}(1 - \frac{(1-P_0)^n}{n!}) - (\sum_{d=0}^{n-2} \frac{(n-d)!}{n!} \cdot \frac{(1-P_0)^d}{P_0^{n-1}}) + (\frac{1}{P_0^{n-1}} \cdot \frac{P_0 - P_0^n}{1-P_0}), \tag{3.24}$$

where $n = |V|$ is the number of vertices in G.

Proof. Theorem 3.7.5 can be proved using the similar procedure that proves Theorem 2 in [59]. □

With the approximate expected number, we can set the stopping criteria to terminate the two-phase algorithm under certain probability significance. We utilize the expected number in Eq. (3.24) and apply the Markov inequality: $P(|Y| \geq \varepsilon) \leq \frac{1}{\varepsilon^{\alpha}}\mathbf{E}(|Y|)^{\alpha}$ to get the upper bound of the number of iterations for finding the optimal solution for DSA.

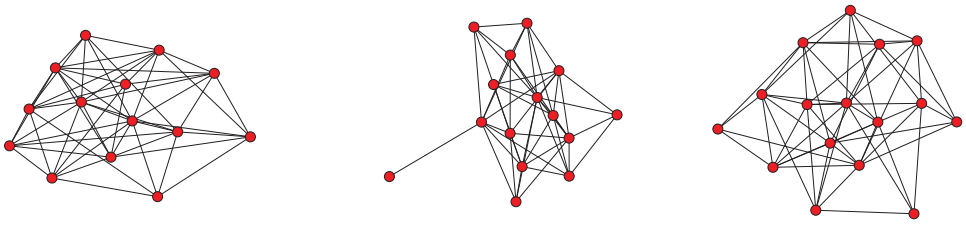
3.8 Numerical Results

In this section, we evaluate the performance of our proposed two-phase algorithm. As DSA is a new spectrum assignment model, there is no existing heuristic algorithm for comparison. Hence, we applied Pure Random Algorithm (PRA) as the benchmark algorithm, in which we randomly selected a vertex order at each iteration and calculate the optimal solution for this vertex order by using *Algorithm* 3.2. The ILP model for DSA was solved by MATLAB2015a with the CPLEX toolbox and the approximate solutions from our two-phase algorithm and PRA were both obtained with MATLAB2015a under the same number of iterations. We run 30 independent simulations on each conflict graph and average the results to ensure sufficient statistical accuracy. We set the probability of moving in the correct direction as $P_0 = 0.5$ in Eq. (3.24) and the significance probability as 90%. All the simulations run on a computer with 3.2 GHz Intel(R) Core(TM) i5-4690S CPU and 8 GBytes RAM.

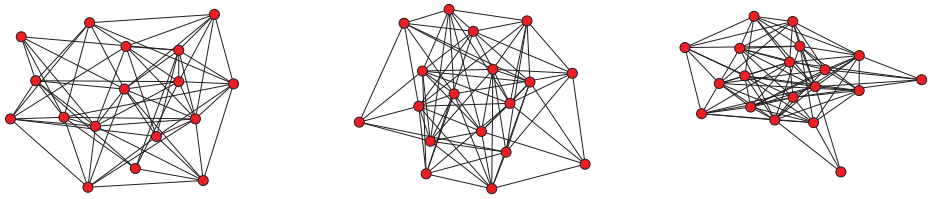
3.8.1 Simulation Setup

We perform simulations in different scenarios:

- **Random graphs**: We use the NetworkX package [3] to generate random graphs, in which each vertex pair is directly connected with a probability of 0.5, as DSA conflict graphs. The weights of vertices and edges are randomly chosen within $[1, |V|]$. Specifically, Fig. 3.9 shows the six random graphs that are used in the simulations. They have $|V| \in [14, 19]$. Hence, we assessed the performance of *Algorithm* 3.3 and our two-phase algorithm under the pure random conditions.
- **Complete graphs with random weights**: To reveal the effectiveness of the two-phase algorithm, we also use complete conflict graphs with $|V| \in [14, 19]$, whose vertex and edge weights were also randomly chosen within $[1, |V|]$, as the DSA conflict graphs.
- **Edge number**: By intuition, the more edges or the larger the biggest guard-band size that a conflict graph has, the bigger its MUFI is. Therefore, we apply the two-phase algorithm on six random conflict graphs, each of which has 14 vertices and the number of edges ranges within {15, 30, 45, 60, 75, 90} as shown in Fig. 3.10. The vertex and edge weights are still chosen randomly as above.

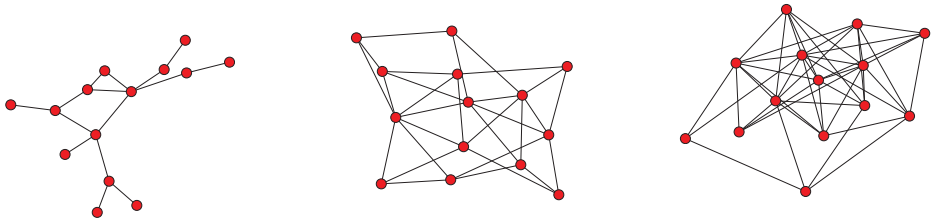


(a) 14-vertex (b) 15-vertex (c) 16-vertex

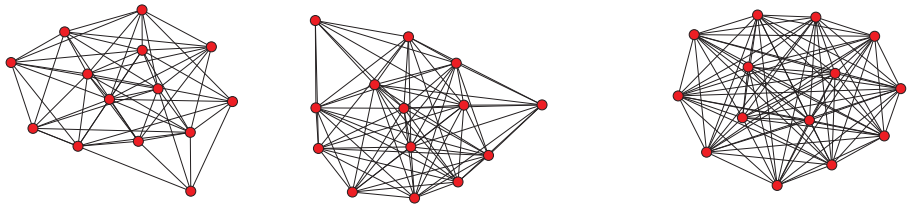


(d) 17-vertex (e) 18-vertex (f) 19-vertex

Figure 3.9: *Six random graphs with 14-19 vertices.*



(a) 15 edges (b) 30 edges (c) 45 edges



(d) 60 edges (e) 75 edges (f) 90 edges

Figure 3.10: *Six random graphs with 14 vertices and 15-90 edges.*

- **14-node NSFNET and 28-node US Backbone**: To mimic the realistic situations, we run simulations on two practical EON topologies, i.e., the 14-node NSFNET and the 28-node US Backbone [29]. Here, each lightpath request is randomly generated and we use the shortest path to route it. The guard-band requirement between two lightpaths is computed as the number of common links on their routing paths. Following these principles, DSA conflict graphs are constructed and we applied the two-phase algorithm to solve the DSA problems.

3.8.2 Simulation Results

Random Graphs

Table 3.3 presents the average MUFI computed by PRA, FPGA, two-phase and ILP-DSA, respectively for the six random topologies in Fig. 3.9. The relative gaps (errors-optimal ratios) with a 95% confidence interval are shown in Fig. 3.11. In Table 3.3, both the initial solutions from FPGA and the improved solutions from the two-phase algorithm are better than those from PRA under the same number of iterations. We also observe that the solutions are truly improved in the second phase, since the MUFI from the two-phase algorithm are closer to the optimal one obtained from FPGA, as shown in Fig. 3.11. Another notable fact is that the results of Fig. 3.9(b) are better than those in Fig. 3.9(a). We observe that there is a vertex with degree one in the topology of Fig. 3.9(b), which is different from Fig. 3.9(a). This fact implies that the topology does have impact on the final MUFI.

Table 3.3: *Numerical Results for Fig. 3.9*

Fig. 3.9	(a)	(b)	(c)	(d)	(e)	(f)
PRA	95.2	92.8	104.	128.	147.	160.
FPGA	75.4	75.3	77.4	96.5	101.	111.
Two-phase	72.7	72.5	75.5	91.0	99.4	110.
ILP-DSA	71.6	70.1	73.6	87.5	94.5	105.

Random Complete Graphs

Table 3.4 presents the average MUFI obtained in the six random complete graphs. The relative gaps with a 95% confidence interval are shown in Fig. 3.12. We can observe the similar trends as discussed above for random conflict graphs. Moreover, we can see that both the relative gaps and the confidence intervals in complete graphs are smaller than those in random graphs for two-phase, FPGA and PRA. This can be interpreted as follows. In complete graphs, the FS set assigned to each vertex should be mutually disjoint, which makes the optimal MUFI (computed by the ILP) bigger. While in random graphs, the FS sets assigned to certain vertices could be overlapped, and hence the optimal value of MUFI would be smaller. However, the overlapped FS sets make

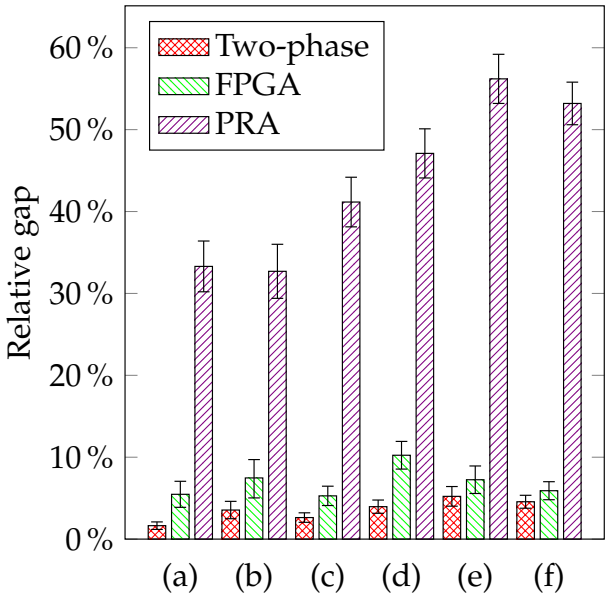


Figure 3.11: *Relative gaps of Table 3.3 by Two-phase, FPGA and PRA.*

it more difficult for the three algorithms to optimize the spectrum assignment, which leads to smaller relative gaps and confidence intervals in complete graphs.

Table 3.4: *Numerical Results for Random complete graphs*

# vertices	14	15	16	17	18	19
PRA	169.0	194.1	238.7	259.1	283.3	297.3
FPGA	145.0	164.7	197.4	216.5	234.4	246.1
Two-phase	143.6	163.6	196.6	213.3	231.3	241.5
ILP-DSA	142.4	160.5	191.1	207.6	223.7	231.8

Edge number

Fig. 3.13 plots the simulation results on six random graphs in Fig. 3.10. The results on MUFI from two-phase algorithm and ILP-DSA are marked as purple and blue bars respectively, and the approximate ratio is plotted in red line. It can be seen that the approximate ratio of the two-phase algorithm increases with the number of edges in the conflict graph.

These results coincided well with the intuitive observation that the more edges or the bigger edge weights that a graph has, the more spectrum resources that DSA would consume. The feature also inspires us that a good routing algorithm should be used to reduce the common links and thus further improve the quality of the results for DSA in EONs.

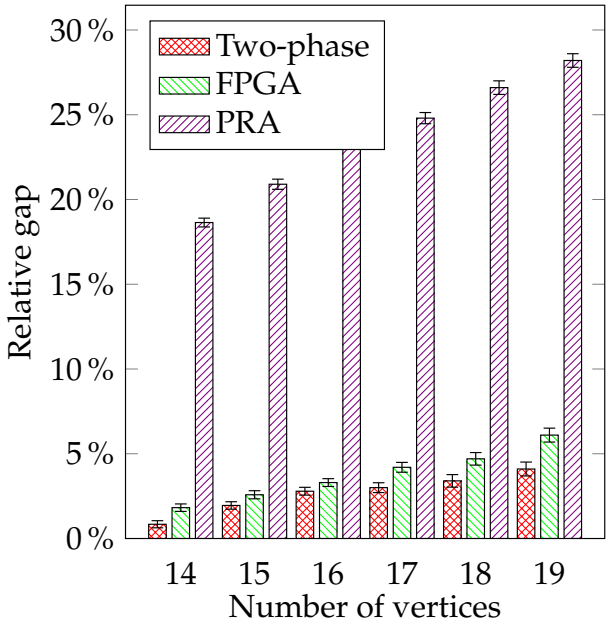


Figure 3.12: *Relative gaps of Table 3.4 by Two-phase, FPGA and PRA.*

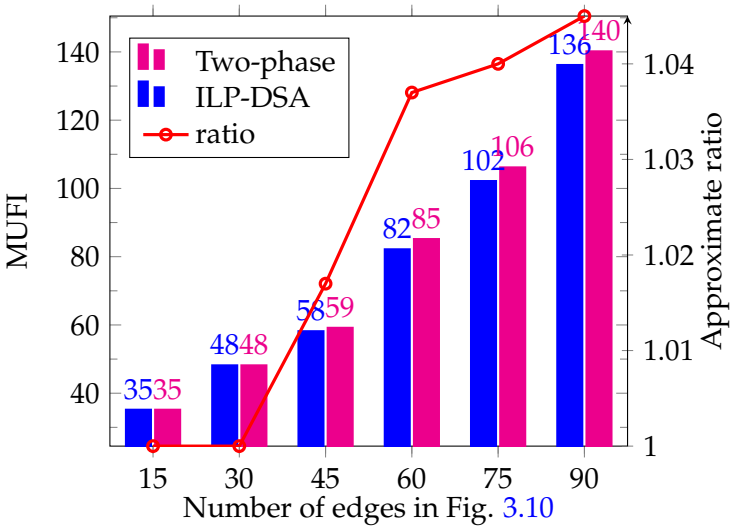


Figure 3.13: *Numerical results for Edge number scenario.*

Table 3.5: *Simulation Results for EONs with NSFNET and US Backbone Topologies*

	# requests	**ILP-DSA**	**Two-phase**	**PRA**		# requests	**ILP-DSA**	**Two-phase**	**PRA**
	10	29	29	29		10	33	33	33
	20	72	72	76		30	186	189	197
NSFNET	30	153	153	177	US	50	351	363	462
	40	200	201	252		100	—	1339	1898
	50	420	423	500	Backbone	150	—	2843	4666
	60	—	469	602		200	—	3784	7743
	70	—	598	805		250	—	6347	13020
	80	—	890	1155		300	—	8140	17303

14-vertex NSFNET and the 28-vertex US Backbone

We evaluate the performance of two-phase algorithm with two practical EON topologies. In Table 3.5, we can see that ILP-DSA can only get the optimal solution when the number of lightpaths is within 50. Meanwhile, our two-phase algorithm can obtain almost the same solutions as ILP-DSA.

Based on all these observations, we can conclude that our proposed two-phase algorithm can approximate the optimal solution for DSA well.

3.9 Conclusions

In this chapter, we studied the DSA problem in EONs. By reducing MHP and graph coloring to DSA, we have proven that DSA is $\mathcal{NP}$-hard and inapproximable. Then, we analyzed and provided the upper and lower bounds for the optimal solutions of DSA, and proved that they are tight. Next, by leveraging a vertex order and developing a polynomial-time algorithm (*i.e.*, *Algorithm* 3.2), we transformed DSA into POP. Then, we developed a two-phase algorithm to solve DSA time-efficiently. For the first phase (*i.e.*, *Algorithm* 3.3) in the algorithm, we theoretically proved that its time complexity is $\mathcal{O}(|V|^3 \cdot \Delta)$, and it can get the optimal solution for bipartite conflict graphs and guarantee an approximate ratio of $\mathcal{O}(\log(|V|))$ for complete conflict graphs with triangle inequality. The second phase utilized a random optimization algorithm, and we applied theoretical analysis to obtain the expected number of iterations for getting the optimal solution. The numerical simulation results demonstrated that our two-phase algorithm can find the near-optimal solutions for DSA in various conflict graphs.

Chapter 4

Virtual Network Embedding: Paths and Cycles

Contents

4.1 Introduction

In previous two chapters, we investigated the RSA problem in EONs from lightpath routing to spectrum assignment. In this chapter, we turn our attention to another promising technology in next generation communication networks, network virtualization. As introduced in Chapter 1, the diversification of current Internet infrastructures results in a serious Internet ossification problem [67], and Network virtualization has been proposed to overcome the Internet ossification and attracting a lot of researches [10, 17, 25, 28, 33, 77]. It supports various networks of diverse natures (*e.g.*, network architectures, protocols, and user interactions [77]) to coexist in a same substrate network and share substrate resources (*e.g.*, CPUs and bandwidths). In the paradigm of network virtualization, the role of traditional Internet Service Providers (ISPs) is separated into two new entities: Infrastructure Provider (InP) and Service Provider (SP). The InP owns and manages the substrate network while the SP focuses on offering customized services to clients. In this business model as shown in Fig. 4.1, the InP sets up and maintains the physical equipments and substrate resources such as optical fibers, CPUs and bandwidths as well as network protocols. Herein, these physical equipments and resources compose the attributes of the InP, which serve to discover resources for SPs [10]. The SP, pursuant to service demands of clients, creates Virtual Network Requests (VNR) (A VNR is a combination of Virtual Nodes (VNs) and Virtual Links (VLs) [25]). It then discovers resources available in substrate networks by the attributes of InPs and selects appropriate ones for the deployment of VNRs [10].

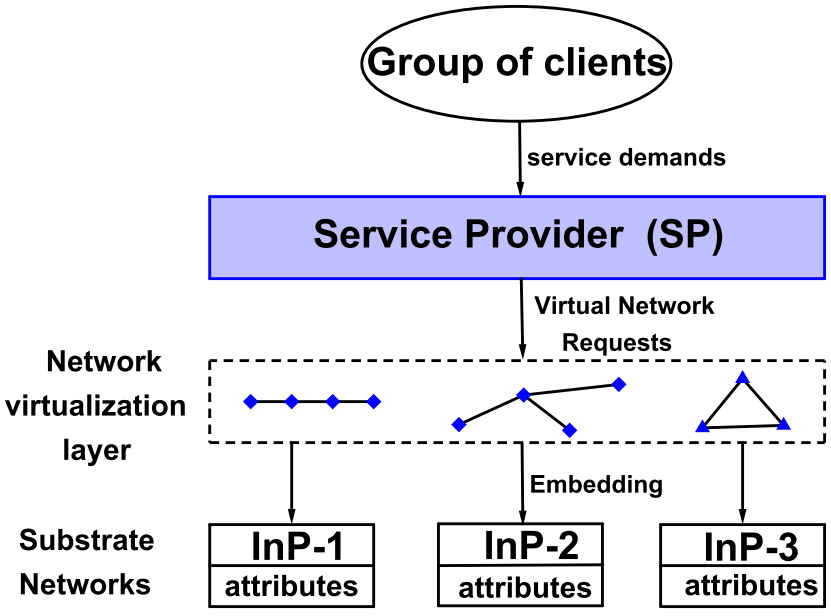


Figure 4.1: *The paradigm of network virtualization.*

How to effectively allocate resources of the substrate network to VNRs is a vital problem in network virtualization, which is often referred to as the Virtual Network

Embedding (VNE) problem [25]. Explicitly, the VNE needs to (a) find a Substrate Node (SN) to meet the computing requirement of each VN, and (b) find a substrate path to satisfy the bandwidth requirement of each VL in a VNR. The former is also called *Node Mapping* and the latter is named *Link Mapping*. The VNE has been proven $\mathcal{NP}$-hard [4] and studied intensively [17, 19, 28, 33, 34, 45, 51, 77, 81]. These works introduce different methods like heuristic algorithms and Integer Linear Programming (ILP) models, *etc*, and cover many aspects, such as distributed computing of the VNE and embedding across multiple substrate networks.

One of the key impediments in the VNE problem is the topological heterogeneity of both VNRs and substrate networks [10]. However, this is not always true in many specific applications and substrate networks. For instance, the topologies of network service chains are paths [38], and there are many substrate optical rings (*i.e.*, cycles) [61]. For these applications and infrastructures, specialized cloud service providers outperforming the general SPs are desired, where dedicated algorithms, taking into account the topological characteristics of the VNRs and substrate networks, can be afforded. Besides, paths and cycles are two of the most fundamental topologies in network structures. Exploiting the characteristics of path and cycle embeddings is vital to tackle general topology embedding.

For example, if path and cycle embeddings can be effectively solved, we can decompose a general VNR into paths and cycles and then embed them on the specialized platforms, as shown in Fig. 4.2, to boost the performance of the VNE. (The feasibility of embedding a VNR across multiple substrate networks has been verified in [33], which makes the idea of decomposition practicable.)

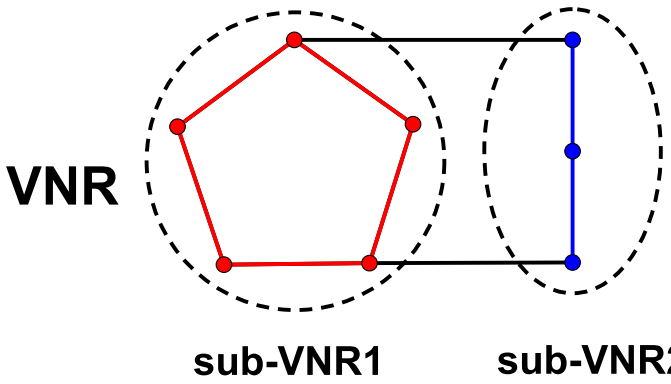


Figure 4.2: *A general VNR decomposed into two sub-VNRs, one cycle and one path.*

In the wake of the idea of the two special embeddings (paths and cycles), there are some important questions not answered yet: How hard are they? Still $\mathcal{NP}$-hard or there exist polynomial-time algorithms to solve them exactly or approximately? Which kind of substrate network topology is more suitable for the special embeddings? In this chapter, we comprehensively investigate path and cycle embeddings from hardness and approximation analysis to effective-algorithm design. The main contributions of this work are summarized as follows:

- We proved that path embedding problem is still $\mathcal{NP}$-hard even in a simplified model. Leveraging Eulerian trail, some approximation algorithms are thus proposed for the first time. We further characterized the topologies of substrate networks which is more suitable for path embedding.
- For path embedding in realistic scenarios, we gave the inapproximability result of path embedding. By transforming this problem into a Multiple Knapsack Problem (MKP) and Multi-Dimensional Knapsack Problem (MDKP), efficient and effective MKP-MDKP-based algorithms are developed.
- For cycle embedding, we proposed a Weighted Directed Auxiliary Graph (WDAG) and succeeded to establish a one-to-one relation between each directed cycle in WDAG and each feasible embedding. Based on that, a polynomial-time algorithm is herein devised to achieve the least-resource-consuming embedding.

The rest of this chapter is organized as follows. Section 4.2 briefly introduces the related work and our motivation. We present the network models and the formal description of the VNE problem in Section 4.3. Then for path embedding, in our preliminary model, the proof of $\mathcal{NP}$-hardness and some approximation algorithms are provided in Section 4.4. For realistic scenarios, we present the inapproximability result and devise the MKP-MDKP-based algorithms in Section 4.5. For cycle embedding, Section 4.6 elaborates the construction of WDAG, characterizes the one-to-one relation between directed cycles and feasible embedding ways, and further devise the specialized cycle-embedding algorithm. We conduct simulations under different scenarios in Section 4.7 to demonstrate the superiority of our proposed algorithms over the existing general algorithms in the two special embeddings. Finally, Section 4.8 summarizes this paper.

4.2 Related Work and Motivation

The VNE, as the main challenging problem in network virtualization, drew a lot of attentions of researchers. In [25], the authors expanded the roles of the SP and InP in the paradigm of network virtualization and proposed a novel classification scheme for current VNE algorithms. Another comprehensive survey [10] elaborated and emphasized the importance of resource discovery and allocation of the VNE. Many solutions to the VNE problem have been proposed in the literature [17, 19, 28, 33, 34, 45, 51, 77, 81] including heuristic-based, ILP, *etc*. Later in [17, 77], researchers found that the topology information of VNRs and substrate networks can be utilized to improve the performance of the VNE. The authors of [17] applied a Markov random walk model, analogous to the idea of PageRank [14], to rank network nodes based on its resource and topological attributes. In [77], customized embedding algorithms for some special classes of topologies have been investigated and proven more effective than the general algorithms. Although VNRs may have arbitrary topologies, the network structures of some key applications are of common topologies *e.g.*, paths and cycles [38, 61]. However, few relevant works intentionally pay attention to the two special but relatively

common topologies in the VNE problem. Besides, paths and cycles are two of the most fundamental topologies in network structures. Since the general VNE problem is computationally hard, it is a pragmatic way to decompose VNRs into several specific substructures of paths and cycles and then effectively embed them separately. Embedding across multiple substrate networks and distributed embeddings of the VNE have been shown to be feasible [25, 33]. This makes the idea of decomposing VNRs and embedding separately practicable. Therefore, a devoted study to explore the characteristics of path and cycle embeddings is desired, which has not yet been researched.

In this chapter, we shall systematically, from theoretical hardness analysis to practical algorithm design, investigate the VNE problem for the two special topologies.

4.3 Network Models and Problem Description

In this section, we will present the network models considered in this chapter and the formulation of the VNE. Some necessary notations are summarized in Table 4.1.

4.3.1 Network Models

Substrate Network

In this chapter, the substrate network $G^s(V^s, E^s)$ is an undirected connected graph. Usually, there are kinds of resources and attributes on the substrate network managed by the InP, such as the computing capabilities and attributes (*e.g.* locations, storages and protocols *etc*) of the SNs and bandwidths of the SLs. Here, we denote by $CPU(v^s)$ and $\mathcal{A}(v^s)$ the computing capability and the attribute array of each SN v^s, and denote $BW(e^s)$ as the bandwidth of SL e^s. Figure 4.3 gives an example of a 4-node substrate network, where each number in the square indicates the CPU capability of the SN and each number beside the SL indicates its BW (The attribute arrays $\mathcal{A}(v^s)$ are not shown explicitly).

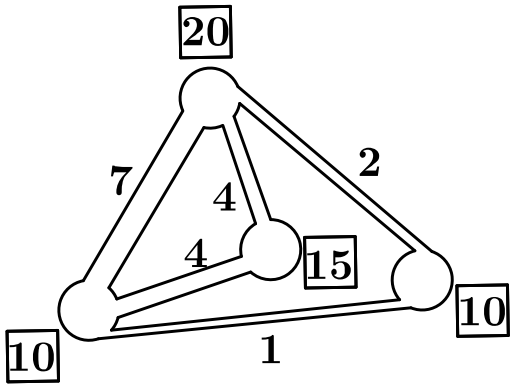


Figure 4.3: *A 4-node substrate network.*

Table 4.1: *Notations of Chapter 4*

Notation	Description
$G^s(V^s, E^s)$	A substrate network, where V^s is the set of SNs, and E^s is the set of Substrate Links (SLs)
v^s	$v^s \in V^s$, a Substrate Node (SN)
e^s	$e^s \in E^s$, a Substrate Link (SL)
$v_i^s v_j^s$	$v_i^s v_j^s \in E^s$, the SL connecting $v_i^s \in V^s$ and $v_j^s \in V^s$
$deg(v^s)$	The degree of v^s, *i.e.*, the number of SLs e^s connecting with v^s
$o(G^s)$	The number of SNs $v^s \in V^s$ of odd degrees
$\mathcal{P}_{v_i^s v_j^s}$	The set of all substrate paths from v_i^s to v_j^s in G^s
$P_{v_i^s v_j^s}$	$P_{v_i^s v_j^s} \in \mathcal{P}_{v_i^s v_j^s}$, a substrate path from v_i^s to v_j^s
P^s	A substrate network which is a path with $\|P^s\|$ SLs on it
$C^s(v_1^s, ..., v_m^s v_1^s)$	A substrate cycle with clockwise order of SNs *i.e.*, starting from v_1^s clockwise to v_m^s, where m is the number of SNs on C^s
$G^r(V^r, E^r)$	A Virtual Network Request (VNR), where V^r is the set of VNs and E^r is the set of VLs
v^r	$v^r \in V^r$, a Virtual Node (VN)
e^r	$e^r \in E^r$, a Virtual Link (VL)
$v_i^r v_j^r$	$v_i^r v_j^r \in E^r$, the VL connecting $v_i^r \in V^r$ and $v_j^r \in V^r$
$P^r(v_1^r v_2^r ... v_n^r)$	A path VNR with n VNs, *i.e.*, from v_1^r, through $v_{j,2\leq j\leq n-1}^r$, to v_n^r.
$\|P^r\|$	The length of P^r, *i.e.*, the number of VLs in P^r
$C^r(v_1^r, ..., v_n^r v_1^r)$	A cycle VNR with clockwise order of VNs, where n is the number of VNs on C^r
$CPU(v^s)$	The CPU capacity of SN v^s
$\mathcal{A}(v^s)$	The array of attributes of SN v^s, such as the location, storage, protocol *etc.*
$BW(e^s)$ or $BW(v_i^s v_j^s)$	The bandwidth capacity of SL e^s or $v_i^s v_j^s$
$CPU(v^r)$	The CPU demand of VN v^r
$\mathcal{A}(v^r)$	The array of attribute requirements of VN v^r, such as the location, storage, protocol *etc.*
$\Phi(v^r)$	The set of SNs v^s that $\mathcal{A}(v^s)$ fit $\mathcal{A}(v^r)$ denoted by $\{v^s \in V^s \| \mathcal{A}(v^r) \to \mathcal{A}(v^s)\}$.
$BW(e^r)$ or $BW(v_i^r v_j^r)$	The bandwidth demand of SL e^r or $v_i^r v_j^r$
$\mathcal{F}_{v^r}$	$\{v^s \in V^s \| CPU(v^s) \geq CPU(v^r)\}$, *i.e.*, the set of feasible SNs whose CPUs are not smaller than VN v^r's
$\mathcal{F}_{e^r}$ or $\mathcal{F}_{v_i^r v_j^r}$	$\{e^s \in E^s \| BW(e^s) \geq BW(e^r)$ or $BW(v_i^r v_j^r)\}$, *i.e.*, the set of feasible SLs whose bandwidths are not smaller than e^r or $v_i^r v_j^r$
$v^r \to v^s$	v^r is embedded on v^s
$e^r \to e^s$	The embedded substrate path of e^r which passes through e^s

Virtual Network Request

A VNR $G^r(V^r, E^r)$ is still modeled by an undirected connected graph, which is constructed by the SP according to the service demands of clients. The demanded computing capability of each $v^r \in V^r$ is $CPU(v^r)$. In some studies such as Location-constrained VNE (LC-VNE) [28], the VNR has location constraints that some VNs can only be embedded onto specific locations of SNs. For this kind of VNRs, besides the computing capability, there exists an array of attribute requirements $\mathcal{A}(v^r)$ on each v^r such that the v^r can only be embedded onto a subset of V^s: $\Phi(v^r) = \{v^s \in V^s | \mathcal{A}(v^r) \rightarrow \mathcal{A}(v^s)\}$, where $\mathcal{A}(v^s)$ satisfies the requirements of $\mathcal{A}(v^r)$. The demanded bandwidth of each $e^r \in E^r$ is $BW(e^r)$. In this work, we focus on two special topologies of VNRs *i.e.*, paths and cycles. Figure 4.4 illustrates a path VNR and a cycle VNR, where, similarly, the numbers in squares (beside VLs respectively) indicate the corresponding CPUs of VNs (BWs of VLs respectively).

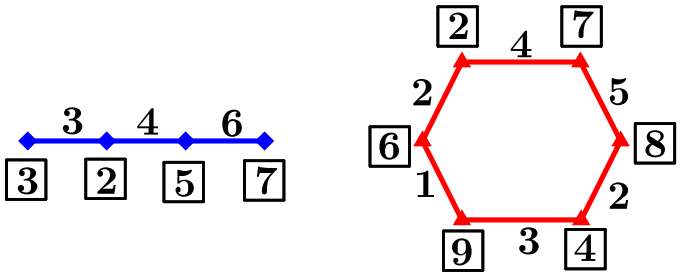


Figure 4.4: *An example of path and cycle VNRs.*

The scope of this paper

In the study of the VNE problem, many literatures [17, 73, 81] focus on the generic setting VNRs, where only the CPU requirement of each VN is taken into account. There are also some works [28] considering special setting VNRs, where an array of attribute requirements $\mathcal{A}(v^r)$ is imposed by the VNR. In this chapter, we call the former as G-VNE and the latter as S-VNE. From the theoretical point of view, the G-VNE is in fact a special case of the S-VNE by setting each $\Phi(v^r) = V^s$, which means that all SNs v^s are eligible to embed the VN v^r. **In this chapter, we mainly focus on the G-VNE problem.** Our work addresses two aspects of the VNE problem in paths and cycles as follows.

- Hardnesses Analysis: Since the G-VNE is a special case of the S-VNE, all of the conclusions about the $\mathcal{NP}$-hardnesses and inapproximability of the G-VNE in paths and cycles can be directly applied to the S-VNE with the same topology.
- Heuristic Design: As we shall see, the VNE problem is still extremely hard even in paths and cycles. Effective heuristics, which capture the "nature" of the problem to some extent, are thus more practical to solve the VNE problem.

4.3.2 Problem Formulation

Constraints

As mentioned above, the VNE problem contains two constraints: *Node Mapping Constraint* and *Link Mapping Constraint*.

Node Mapping Constraint For the node mapping, there are two main embedding methods in the literature: *Many-to-One* and *One-to-One* [28].

- *Many-to-One* (M2O): In this type of node embedding, given the $G^s(V^s, E^s)$ and a VNR $G^r(V^r, E^r)$, each VN $v^r \in V^r$ must be embedded onto such an SN $v^s \in V^s$, *i.e.*, $v^r \rightarrow v^s$, that $CPU(v^s) \geq CPU(v^r)$. Meanwhile, for each SN v^s on which some VNs $v^r \in V^r$ are embedded, $CPU(v^s) \geq \sum_{v^r \rightarrow v^s} CPU(v^r)$. Figure 4.5 illustrates an example of the M2O embedding.

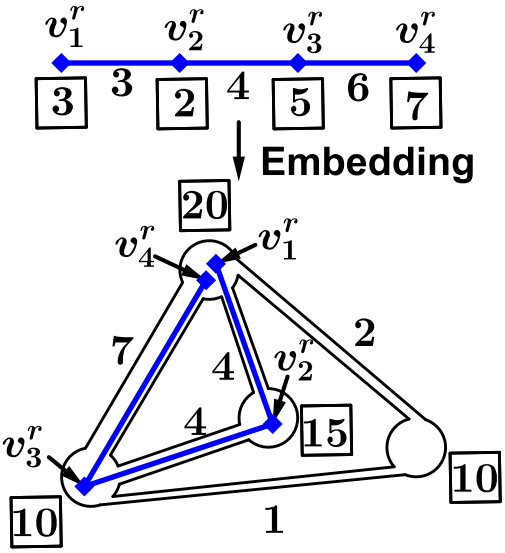


Figure 4.5: *Many-to-One (M2O) node embedding.*

- *One-to-One* (O2O): As the M2O embedding sometimes makes a VNR more vulnerable to the substrate network failure [28], some literatures add an additional constraint that two VNs v_i^r and v_j^r of a same VNR can not be embedded on a same SN v^s, *i.e.*, if $v_1^r \rightarrow v^s$ and $v_2^r \rightarrow v^s$ then $v_1^r = v_2^r$. Thus, the embedding way of v_1^r and v_4^r in Fig. 4.5 is not allowed in the O2O embedding.

Link Mapping Constraint For each VL $v_i^r v_j^r$, assuming $v_i^r \rightarrow v_i^s$ and $v_j^r \rightarrow v_j^s$, $v_i^r v_j^r$ should be embedded on a substrate path $P_{v_i^s v_j^s} \in \mathcal{P}_{v_i^s v_j^s}$, and for each SL e^s on $P_{v_i^s v_j^s}$, $BW(e^s) \geq BW(v_i^r v_j^r)$. Meanwhile, for each SL e^s, through which the embedded sub-

strate paths of VLs e^r pass, *i.e.*, $e^r \rightarrow e^s$, $BW(e^s) \geq \sum_{e^r \rightarrow e^s} BW(e^r)$. In this chapter, the VNs embedded on a same SN are isolated *i.e.*, the SN does not support intercommunications between VNs [70].

Here, we give the definition of the decision problem $Emb(\cdot, \cdot)$ as follows.
Definition 4.3.1. *$Emb(G^s, G^r)$ is such a decision problem that its answer is **Yes** iff the VNR G^r can be embedded on the substrate network G^s satisfying the node and link mapping constraints, and **No** otherwise.*

The definition can be extended to a set of n VNRs: $Emb(G^s, \{G_1^r, G_2^r, ..., G_n^r\})$ whose answer is **Yes** iff all of $G_{i,1\leq i\leq n}^r(V_i^r, E_i^r)$ can be embedded on the G^s while simultaneously satisfying the node and link mapping constraints, and **No** otherwise. Here, the node mapping constraint of a set of VNRs embedded on a substrate network is that for each SN v^s, $CPU(v^s) \geq \sum_{v^r \rightarrow v^s} CPU(v^r), \forall v^r \in V_1^r \cup V_2^r \cup ... \cup V_n^r$, and the link constraint means for each SL e^s, $BW(e^s) \geq \sum_{e^r \rightarrow e^s} BW(e^r), \forall e^r \in E_1^r \cup E_2^r \cup ... \cup E_n^r$. Figure 4.6 shows two path VNRs embedded on the 4-node substrate network in the M2O embedding.

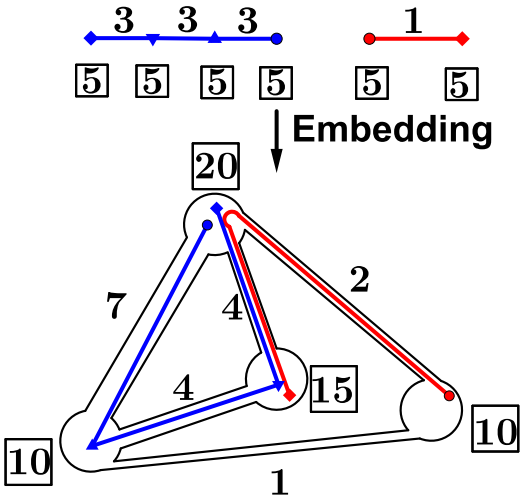


Figure 4.6: *Two path VNRs embedded on a 4-node substrate network.*

Objective Functions

There are two main objective functions in the study of the VNE problem: The acceptance ratio and revenue.

The Acceptance Ratio (AcR) Given a substrate network G^s and a set of VNRs to be served on G^s $\{G_1^r, G_2^r, ..., G_n^r\}$, the objective is to maximize the number of VNRs that can be embedded on the G^s. We denote by **AcR** the acceptance ratio problem as formally defined below.

$$\begin{aligned} &\text{Maximize} \quad |S| \quad (\textbf{AcR}), \\ &\textbf{\textit{s.t.}} \quad Emb(G^s, S) \text{ is } \textbf{Yes}, \\ &\quad S \subseteq \{G_1^r, G_2^r, ..., G_n^r\}. \end{aligned} \tag{4.1}$$

The Revenue (Rev) Each VNR G_i^r is associated with a revenue w_i. Its objective is to maximize the total revenue of VNRs that can be embedded on G^s. We denote by **Rev** the revenue problem defined below.

$$\begin{aligned} &\text{Maximize} \quad \sum_{G_i^r \in S} w_i \quad (\textbf{Rev}), \\ &\textbf{\textit{s.t.}} \quad Emb(G^s, S) \text{ is } \textbf{Yes}, \\ &\quad S \subseteq \{G_1^r, G_2^r, ..., G_n^r\}. \end{aligned} \tag{4.2}$$

One may notice that AcR actually is a special case of Rev by setting each revenue w_i to be one. In this chapter, the set of VNRs $\{G_1^r, G_2^r, ..., G_n^r\}$ is particularly a path set or a cycle set, *i.e.*, $\{P_1^r, P_2^r, ..., P_n^r\}$ or $\{C_1^r, C_2^r, ..., C_n^r\}$.

4.4 Path Embedding in the Preliminary Model

In this section, we will explore the path embedding problem in a preliminary model. This helps shed light on the characteristics of embedding for this special type of topologies and inspires us to develop efficient and effective algorithms for path embedding in realistic scenarios.

The preliminary model is defined with the following network configurations: ① The substrate network $G^s(V^s, E^s)$ is that: $\forall e^s \in E^s$, $BW(e^s) = 1$; $\forall v^s \in V^s$, $CPU(v^s) = deg(v^s)$, where $deg(v^s)$ is the degree of v^s. ② The path VNR $P^r(v_1^r v_2^r ... v_n^r)$ is that: $\forall i$, $CPU(v_i^r) = 1$; $\forall v_i^r v_{i+1}^r$, $BW(v_i^r v_{i+1}^r) = 1$. This preliminary model can be interpreted as follows: Each SL e^s, whose bandwidth is 1 unit, can be utilized by at most one VL e^r, and the computing capability of each SN v_i^s $CPU(v_i^s) = \sum_{v_i^s v_j^s \in E^s} BW(v_i^s v_j^s)$. **All the substrate networks and VNRs discussed in this section are in the framework of the preliminary model and G-VNE.**

4.4.1 The Hardnesses

In path embedding, given a substrate network G^s and a path VNR P^r, an elementary and essential question is how hard $Emb(G^s, P^r)$ is. Since there are two node embedding modes, the O2O and the M2O, we inspect the hardness of $Emb(G^s, P^r)$ in each mode respectively.

$Emb(G^s, P^r)$ problem of the O2O embedding

Let $G^s(V^s, E^s)$ and P^r be the substrate network and a path VNR respectively. In the O2O model, the number of VNs is at most $|V^s|$. Thus, a special case of $Emb(G^s, P^r)$ is to answer whether a path VNR $P^r(v_1^r v_2^r, ..., v_{|V^s|}^r)$ can be embedded on $G^s(V^s, E^s)$. The following lemma gives the necessary and sufficient condition that whether a P^r of $|V^s|$ VNs can be embedded on G^s in the O2O model.

Lemma 4.4.1. *In the preliminary model with the O2O embedding, a P^r of $|V^s|$ VNs can be embedded on the G^s **if and only if** G^s has a trail traversing all SNs. In other words, G^s contains a spanning subgraph having an Eulerian trail.*

Proof. If $P^r(v_1^r v_2^r, ..., v_{|V^s|}^r)$ can be embedded on G^s by, *w.l.o.g.*, $v_i^r \rightarrow v_i^s, \forall i$, then we can express this embedding way as a sequence "$v_1^s P_{v_1^s v_2^s} v_2^s P_{v_1^s v_3^s} ... v_{|V^s|}^s$". This sequence, if expressing each $P_{v_i^s v_{i+1}^s}$ by its vertex sequence, in fact is a trail which traverses all SNs in G^s (but not necessary all SLs).

Conversely, if G^s has a trail, say T, traversing all SNs, let, *w.l.o.g.*, "$v_1^s, ..., v_2^s, ..., v_i^s, ..., v_{|V^s|}^s$" be T's corresponding vertex sequence. By the definition of trail, there must be a path connecting v_i^s and v_{i+1}^s, say $P_{v_i^s v_{i+1}^s}, \forall i$, and $P_{v_i^s v_{i+1}^s} \cap P_{v_j^s v_{j+1}^s} = \emptyset, \forall i \neq j$. Therefore we can embed P^r by "$v_1^s P_{v_1^s v_2^s} v_2^s P_{v_1^s v_3^s} ... v_{|V^s|}^s$". □

Lemma 4.4.2. *The hardness of determining whether a graph contains a spanning subgraph having an Eulerian trail (or SSET in short) is equivalent*[1] *to that of determining whether a graph is a Supereulerian graph (or SG).*

Consequently, by Lemmas 4.4.1 and 4.4.2, we have Theorem 4.4.1.

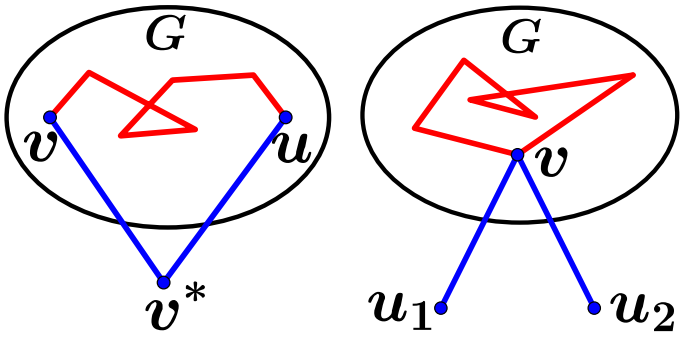
Theorem 4.4.1. *The hardness of $Emb(G^s, P^r)$ problem in O2O embedding is not less than that of determining whether G^s is a Supereulerian graph.*

Proof. To this end, we need to prove that one of the two problems can be reduced to the other in polynomial time.

First, we prove $SSET \leq_T^P SG$: Given a graph $G(V, E)$, for each vertex pair $v, u \in V$, we construct a graph G_{vu} by adding a new vertex v^* and connecting v^*v and v^*u as shown in Fig. 4.7(a). Afterwards, we obtain a set $\mathcal{G} = \{G_{vu} | \forall v, u \in V\}$ and $|\mathcal{G}| = \binom{|V|}{2}$. It is easy to see that G contains a spanning subgraph which has an Eulerian trail iff there exists one $G_{vu} \in \mathcal{G}$ which is a Supereulerian graph.

Then for $SG \leq_T^P SSET$: Given a graph $G(V, E)$, arbitrarily selecting one vertex $v \in V$, we construct a graph G^* by adding two new vertices u_1 and u_2 and connecting them to v as shown in Fig. 4.7(b). It is easy to see that G is a Supereulerian graph iff G^* contains a spanning subgraph which has an Eulerian trail. □

[1] The equivalence is under the polynomial-time Turing reduction.



(a) G_{vu}. (b) G^*.

Figure 4.7: *Polynomial-time reduction of SSET and SG.*

Unfortunately, given a graph G, it is $\mathcal{NP}$-hard to determine whether G is a Supereulerian graph [15].

$Emb(G^s, P^r)$ problem of the M2O embedding

Given a $G^s(V^s, E^s)$ and P^r, the number of VNs of P^r is not limited by the $|V^s|$ for the M2O embedding. Hence, the hardness of $Emb(G^s, P^r)$ problem for the M2O embedding is equivalent to determining the length of the longest P^r that can be embedded on G^s. There is a fact as below.

Fact 4.4.1. *In the M2O embedding, given a $G^s(V^s, E^s)$ and P^r in the preliminary model, P^r can be embedded on G^s* ***if and only if*** *there exits a trail T in G^s with a length of $|P^r|$.*

Proof. If $P^r(v_1^r v_2^r, ..., v_{|P^r|+1}^r)$ can be embedded on G^s by, *w.l.o.g.*, "$v_1^s P_{v_1^s v_2^s} v_2^s P_{v_1^s v_3^s} ... v_{|P^r|+1}^s$", we can easily shorten each substrate path to an SL to obtain a trail T with a length of $|P^r|$.

Conversely, let T be a trail of length $|P^r|$ in G^s whose vertex sequence is, *w.l.o.g.*, "$v_1^s, v_2^s, ..., v_{|P^r|+1}^s$". We can embed P^r on T by $v_i^r \to v_i^s, \forall 1 \leq i \leq |P^r| + 1$. Now, we check the feasibility of this embedding. The link mapping constraint is obviously satisfied. For node mapping constraint, it is easy to see that the number of VNs embedded on a same SN v_i^s will not exceed $deg(v_i^s) = CPU(v_i^s)$. □

Hence, in the M2O embedding, the length of the longest P^r that can be embedded on G^s is equal to the length of the longest trail on G^s.

Theorem 4.4.2. *In the M2O embedding, the hardness of $Emb(G^s, P^r)$ problem in the preliminary model is equivalent to determining the length of the longest trail in G^s.*

Unfortunately, the longest trail problem is also an $\mathcal{NP}$-hard problem [50].

After obtaining above results, immediately, we can get the inapproximability of the AcR and Rev problems of path embedding in the preliminary model as follows.

Theorem 4.4.3. *For path embedding in the preliminary model, both the AcR and Rev problems have an $\mathcal{NP}$-hard gap $[\epsilon, 1]$[2], $\forall 0 < \epsilon < 1$, i.e., unless $\mathcal{NP} = \mathcal{P}$, there is no polynomial-time algorithm of an approximation ratio $\frac{1}{\epsilon}$ for both problems.*

Proof. Given an instance of the AcR problem consisting of a G^s and a P^r, if the answer of $Emb(G^s, P^r)$ is **Yes**, then $OPT_{AcR} \geq 1$, otherwise, $OPT_{AcR} < \epsilon$, where OPT_{AcR} is the optimal solution of the AcR. Since $Emb(G^s, P^r)$ is $\mathcal{NP}$-hard, the AcR problem in the preliminary model has an $\mathcal{NP}$-hard gap[3] $[\epsilon, 1]$. For the Rev, as the AcR is its special case, the proof also follows. □

As mentioned above that the G-VNE is a special case of the S-VNE, all of these results can be directly applied to the S-VNE problem in path embedding.

4.4.2 Some Approximation Algorithms

From the above hardness analysis, any polynomial-time algorithm of a constant approximation ratio is implausible for the AcR and Rev of path embedding in the preliminary model. However, in the preliminary model, if G^s itself is also a path, the AcR and Rev problems can be easily solved as shown in Lemma 4.4.3.

Lemma 4.4.3. *In the preliminary model, if the substrate network is a path denoted by P^s, given a set of $\{P_1^r, P_2^r, ..., P_n^r\}$, then the AcR and Rev problems can be solved in polynomial time.*

Proof. We use Fig. 4.8 to demonstrate the substrate path P^s and the set of path VNRs $\{P_1^r, P_2^r, ..., P_n^r\}$. In this special case, we can regard P^s as a "knapsack" with a capacity of $|P^s|$ (edge number), and the n path VNRs $P^r_{i,1\leq i\leq n}$ as n items with a size of $|P^r_{i,1\leq i\leq n}|$ respectively.

The AcR can be easily solved in this way: First arrange the n path VNRs in the increasing order of their sizes, then sequentially pack them into the "knapsack" until cannot (This embedding is feasible since each middle SN v^s of $CPU(v^s) = deg(v^s) = 2$, and at most two VNs will be embedded on v^s). The Rev problem in this special case is equivalent to the classical 0-1 Knapsack Problem (KP) and it thus can be solved by a dynamic programming algorithm of time complexity $\mathcal{O}(n|P^s|)$ [42] [4]. □

But the topologies of substrate networks in reality could not always be perfect path-like structures. Therefore, given a substrate network G^s, we may hope to "expand" it into a "path" in some way as shown in Fig. 4.9, where we "split" v_3 into two new vertices. One may notice that the final "path" we obtained in Fig. 4.9 is literally the

[2] If an optimization problem has an $\mathcal{NP}$-hard gap $[\alpha, \beta]$, then it has no polynomial-time algorithm of an approximation ratio $\frac{\beta}{\alpha}$ unless $\mathcal{NP} = \mathcal{P}$ [22].

[3] For a maximization problem Π and an $\mathcal{NP}$-hard decision problem Λ, if Λ is **Yes** $\Rightarrow OPT_\Pi \geq \beta$ and Λ is **No** $\Rightarrow OPT_\Pi < \alpha$, where OPT_Π is the optimal solution of Π, then Π has an $\mathcal{NP}$-hard gap $[\alpha, \beta]$ [22].

[4] Although KP is $\mathcal{NP}$-hard, for the substrate network, whose space complexity is $|P^s|$ (not $\log(|P^s|)$), the dynamic programming algorithm runs in polynomial time (rather than pseudo-polynomial time).

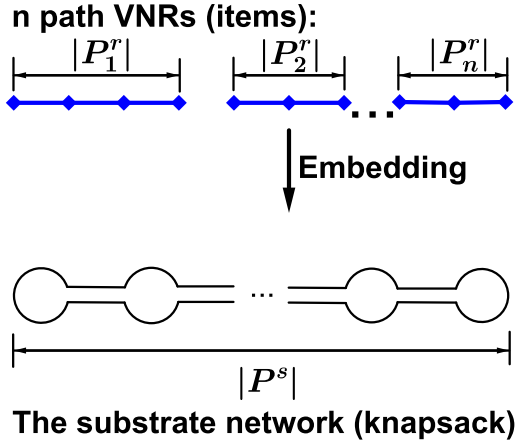


Figure 4.8: *An example of the substrate network being a path.*

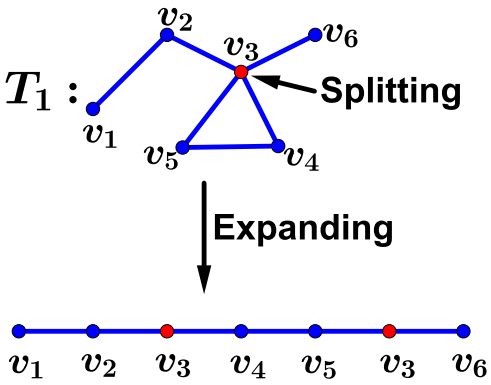


Figure 4.9: *An example of expanding a trail.*

vertex sequence corresponding to the trail T_1. In fact, for all trails, following their vertex sequences, we can expand them into "paths". Given a G^s, the longest trail of $G^s(V^s, E^s)$ is the Eulerian trail (if it has) with a length of $|E^s|$. Here, we give a famous theorem in graph theory which characterizes the type of graphs having an Eulerian trail.

Lemma 4.4.4. *[13] Given a graph G, let $o(G)$ be the number of vertices of odd degree, then G has an Eulerian trail if and only if $o(G) \leq 2$.*

Therefore, for all substrate network G^s with $o(G^s) \leq 2$, we can expand it into a "path" by its Eulerian trail using many polynomial-time algorithms [13]. In the expanded "path", there are several duplicated vertices like the v_3 in Fig. 4.9. For those "split" SNs v^s, since $CPU(v^s) = deg(v^s)$, we dispense $CPU(v^s)$ to its duplicated v'^s in this way: $CPU(v'^s) = 2$ if v'^s is in the middle otherwise, *i.e.*, v'^s is one end of the "path", $CPU(v'^s) = 1$. For example, for the v_3 of T_1 in Fig. 4.9 whose $CPU(v_3) = deg(v_3) = 4$, its two duplicated vertices are of $CPU = 2$. Afterward, we can optimally solve the AcR and Rev problems in the expanded "path" by Lemma 4.4.3. Notice that this method only works for the M2O embedding, because some VNs of a path VNR may be embedded on multiple duplicated vertices of an SN which violates the node mapping constraint

of the O2O embedding.

Theorem 4.4.4. *For the M2O embedding in the preliminary model, given a substrate network G^s and a set of $\{P_1^r, P_2^r, ..., P_n^r\}$, let OPT_{AcR} be the maximum accepted number of VNRs for the AcR problem. There is a polynomial-time algorithm of accepted number APX_{AcR} such that $OPT_{AcR} - APX_{AcR} \leq max(\frac{o(G^s)}{2} - 1, 0)$.*

Proof. If $o(G^s) \leq 2$, we can expand G^s into a "path" by its Eulerian trail and optimally solve the AcR problem by Lemma 4.4.3, and the conclusion follows.

For $o(G^s) > 2$, since $o(G^s)$ is even [13], we can arbitrarily group the vertices of odd degree into $\frac{o(G^s)}{2}$ pairs. From them we arbitrarily select $\frac{o(G^s)}{2} - 1$ pairs and add $\frac{o(G^s)}{2} - 1$ new edges to connect them, resulting in a new graph denoted by G^{s*} with $o(G^{s*}) = 2$. Then, we can expand G^{s*} into a "path" by its Eulerian trail, and solve the AcR problem by Lemma 4.4.3 (not feasible). Finally, we drop such path VNRs embedded on G^{s*} that pass through those newly added edges (now feasible). The number of dropped path VNRs is not greater than $\frac{o(G^s)}{2} - 1$, because each new edge can be utilized by at most one path VNR, and the proof follows. □

For the Rev problem, to develop some approximation algorithms, we may need more informations about the substrate network. Given a substrate network G^s, we can see from the hardness analysis that it is $\mathcal{NP}$-hard to determine whether a path VNR can be embedded on G^s. In other words, whether G^s can serve the path VNR is hard to answer. In fact, the InP should offer its maximum service capacity of a single path VNR in its attributes. When discovering available resources, the SP should not assign the path VNR exceeding the single service capacity to the InP. We call this *MaxCap* business model: The substrate network G^s with a trail T_{max}, and all P^r to be embedded on it with a length of $|P^r| \leq |T_{max}|$.

Theorem 4.4.5. *For the M2O embedding in the preliminary model, given a substrate network G^s with T_{max} and a set of $\{P_1^r, P_2^r, ..., P_n^r\}$, $\forall |P_i^r| \leq |T_{max}|$, there is a polynomial-time algorithm for the Rev problem, which can guarantee an approximation ratio $\rho = max(\frac{o(G^s)}{2}, 1)$.*

Proof. Let OPT_{Rev} be the maximum revenue. For $o(G^s) \leq 2$, we can expand G^s by its Eulerian trail, and the Rev problem corresponds to the case of Lemma 4.4.3 which can be polynomial-timely solved to obtain OPT_{Rev}.

For $o(G^s) > 2$, similarly, we can add $\frac{o(G^s)}{2} - 1$ new edges resulting in a new graph G^{s*} with $o(G^{s*}) = 2$. By G^{s*}'s Eulerian trail, we can obtain a embedding way (not feasible) of a revenue at least OPT_{Rev}. In this embedding way, the number of path VNRs which are embedded on the newly added edges is not bigger than $\frac{o(G^s)}{2} - 1$, and we treat such path VNRs as different parts. Besides, we treat those path VNRs embedded on the original SLs of G^s as one part. Now, we have at most $\frac{o(G^s)}{2}$ different

parts, whose sum of revenue is at least OPT_{Rev}. Therefore, we can select the part of the largest revenue to be embedded on G^s: If the part is a P^r embedded on newly added edges, we can embed it on T_{max} since $|P^r| \leq |T_{max}|$, otherwise the part is those path VNRs already embedded on G^s. Finally the part of the greatest revenue is not less than the average, *i.e.*, $\frac{OPT_{Rev}}{\frac{o(G^s)}{2}}$. □

Moreover, if the InP can offer more information about the substrate network G^s such as providing $\frac{o(G^s)}{2} - 1$ edge-disjoint trails, and the lengths of all P^r to be embedded are not longer than these trails, we can then obtain a better approximation ratio for the substrate network G^s with $o(G^s) > 2$.

Corollary 4.4.1. *For the M2O embedding in the preliminary model, given a G^s with $\frac{o(G^s)}{2} - 1$ edge-disjoint trails where $o(G^s) > 2$, and a set of $\{P_1^r, P_2^r, ..., P_n^r\}$, where $\forall i, |P_i^r|$ is not greater than these trails, there is a polynomial-time algorithm for the Rev problem which can guarantee an approximation ratio $\rho = 2$.*

Proof. Similar to the proof of Theorem 4.4.5, assuming that we have reached the step that there are at most $\frac{o(G^s)}{2}$ parts: at most $\frac{o(G^s)}{2} - 1$ parts are those path VNRs embedded on the newly added edges and the remaining part corresponds to path VNRs embedded on the original SLs. Now, we regard those path VNRs on new edges together as one part, and then we have just two parts that the sum revenue is at least OPT_{Rev}. Hence, our solution is to take the part of the largest revenue to be embedded on G^s. If the part corresponds to the path VNRs embedded on the newly added edges, we embed them on the $\frac{o(G^s)}{2} - 1$ edge-disjoint trails. Otherwise the part corresponds to path VNRs already embedded on G^s. Therefore, the part of the largest revenue is no less than the average $\frac{OPT_{Rev}}{2}$. □

Theorems 4.4.4 and 4.4.5 and Corollary 4.4.1 give us an insight that we can approach the AcR and Rev problems by means of $o(G^s)$. From the above analysis, the smaller $o(G^s)$ a substrate network G^s has, the better approximation ratio the polynomial-time algorithm can achieve. Therefore, these substrate networks with fewer SNs of odd degree are more suitable for path embedding in the preliminary model with the M2O embedding.

4.5 Path Embedding in Realistic Settings

Here, we account for the path embedding problem in the realistic settings, where the *CPU*s of SNs and VNs are arbitrary and so as the *BW*s of SLs and VLs. Thus, in the realistic settings, the path VNE problem should be more difficult than in the preliminary model. Here, we present the following theorem to explicitly show the inapproximability of both the AcR and Rev for path embedding in the realistic settings.

Theorem 4.5.1. *For path embedding in the realistic settings, unless* $\mathcal{NP} \subseteq ZPTIME(n^{polylog(n)})$[5], *there is no* $\mathcal{O}(\log^{\frac{1}{2}-\epsilon}|E^s|)$ *approximation for both the AcR and Rev problems, where* $|E^s|$ *is the number of SLs in the substrate network.*

Proof. First, we introduce the Edge-Disjoint Paths (EDP) problem: Given a connected graph $G(V, E)$ and a set of pairs of vertices $\{(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)\}$ on it, the objective of the EDP is to connect as many pairs as possible via edge-disjoint paths. Unless $\mathcal{NP} \subseteq ZPTIME(n^{polylog(n)})$, there is no $\mathcal{O}(\log^{\frac{1}{2}-\epsilon}|E|)$ approximation for the EDP problem [6].

Now given an instance of the EDP, *i.e.*, a graph $G(V, E)$ where $V = \{v_1, ...v_i, ..., v_n\}$ and a set of pairs of vertices $\{(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)\}$, we translate it into an instance of the AcR in path embedding as follows. To completing the translation, for each $v_i \in V$, we first denote by N_{v_i} the total number of v_i appearing in the sets $\{s_i\}_{i=1}^k$ and $\{t_i\}_{i=1}^k$.

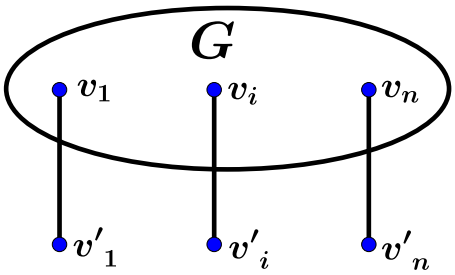


Figure 4.10: *The constructed substrate network.*

The construction of the substrate network G^s is that: The set of SNs V^s consists of two parts, $V = \{v_1, ...v_i, ..., v_n\}$ and $V' = \{v'_1, ..., v'_i, ..., v'_n\}$ where V' is a copy of V. The set of SLs E^s also consists of two parts, E and $\{v_i v'_i\}_{i=1}^n$ as shown in Fig. 4.10.

The setting of *CPUs* and *BWs* is as follows.

- For each $v_i \in V$, $CPU(v_i) = 1$ and for each $e \in E$, $BW(e) = 1$.
- For V', $CPU(v'_1) = C_{v_1} \times N_{v_1}$ where $C_{v_1} = 2$ and for $2 \leq i \leq n$, $CPU(v'_i) = C_{v_i} \times N_{v_i}$ where $C_{v_i} = CPU(v'_{i-1}) + 1$.
- For $\{v_i v'_i\}_{i=1}^n$, $BW(v_n v'_n) = B_{v_n} \times N_{v_n}$ where $B_{v_n} = 2$ and for $n - 1 \geq i \geq 1$, $BW(v_i v'_i) = B_{v_i} \times N_{v_i}$ where $B_{v_i} = BW(v_{i+1} v'_{i+1}) + 1$.

The path VNRs set is $\{P_1^r, ..., P_i^r, ..., P_k^r\}$, where the i-th P_i^r, which corresponds to (s_i, t_i), consists of 4 VNs and its *CPUs* and *BWs* requirements are shown as in Fig. 4.11

From the setting of *CPUs* and *BWs*, we can see that C_{v_i} is increasing with the i growing while B_{v_i} is decreasing. Thus, following the constraints of node and link mappings,

[5] $ZPTIME(n^{polylog(n)})$ is the set of languages that have randomized algorithms that always give the correct answer and have expected running time $n^{polylog(n)}$

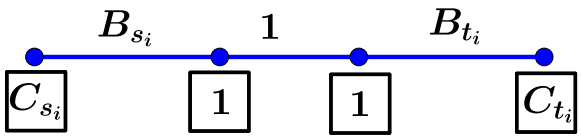


Figure 4.11: *The i-th path VNR P_i^r.*

it is easy to see that the two ends of the P_i^r must be embedded on the copies of s_i and t_i respectively, if the P_i^r is embedded.

Hence, for any pairs we can connect by edge-disjoint paths in the instance of the EDP, we can embed their corresponding path VNRs in the substrate network, vice versa.

Besides, since the number of SLs $|E^s| = |E| + |V| \leq 3|E|$, we get the inapproximability. □

Similarly, this inapproximability result can be directly applied for the S-VNE of path embedding.

Theorem 4.5.1 implies that for path embedding in the realistic settings, it is even implausible to find a polynomial-time algorithms of an approximation ratio with respect to the parameters of the substrate graph like that in the preliminary model. Thus, we should turn our attention from developing approximation algorithms to designing better heuristic algorithms, which is able to capture the "nature" of the path VNE problem.

As proven in the previous section, if G^s is a path, then the AcR and Rev problems can be easily solved in the preliminary model by leveraging KP. This result relatively reflects some essentials of the path VNE problem that can be regarded as "packing" (embedding) a set of "items" (VNRs) into a special "knapsack" (the substrate network). Inspired by this, given a substrate network G^s and a set of $\{P_1^r, P_2^r, ..., P_n^r\}$, we propose a framework of algorithm design for the realistic settings. The main idea is described as follows. First, we decompose G^s into several substrate paths. This phase is thus called path decomposition. By regarding each substrate path as a knapsack and each P_j^r as an item with size $|P_j^r|$ and profit w_j, and we then pack these items into multiple knapsacks, which can be formulated as a Multiple Knapsack problem (MKP). Finally, we assign the CPU and BW resources to those packed path VNRs, and it corresponds to the Multi-Dimensional Knapsack Problem (MDKP). To this end, we review the two well-studied MKP and MDKP.

Multiple Knapsack Problem (MKP)

MKP[42] is a classical variation of KP. In MKP, there are a set of knapsacks $M := \{1, ..., i, ..., m\}$ each with positive capacities b_i, and a set of items $N := \{1, ...j, ..., n\}$ each with size $s_j \geq 0$ and profit $w_j \geq 0$. The goal is to find a subset of the n items of maximum profit which can be packed into the m knapsacks.

In this chapter, we measure the time complexity of solving an instance of MKP by its numbers of knapsacks and items, m and n respectively, denoted by $T_{MKP}(m, n)$.

Multi-Dimensional Knapsack Problem (MDKP)

MDKP[42] is another well-known variation of KP. In d dimensional MDKP denoted by d-DKP, there are a knapsack of d-dimensional positive capacity attributes $(b_1, .., b_i, ..., b_d)$ and a set of items $N := \{1, ...j, ..., n\}$ each with profit w_j and d-dimensional size attributes $(s_{j1}, ...s_{ji}, ..., s_{jd})$, where all of b_i and w_j and s_{ji} are non-negative. The goal is to find a subset of the n items of maximum profit which can be packed into the knapsack while not exceeding each of d-dimensional capacity attributes.

In this chapter, we measure the time complexity of solving an instance of MDKP by its numbers of dimensions d and items n respectively, denoted by $T_{MDKP}(d, n)$. Next, we assume that the substrate network $G^s(V^s, E^s)$ and a set of path VNRs $\{P_1^r, P_2^r, ..., P_n^r\}$ are given as the input of our algorithm.

4.5.1 Path Decomposition Phase

In this phase, we decompose the substrate network into a set of substrate paths. These decomposed paths are treated as the multiple knapsacks for the optimization of the next phase. Intuitively, if these "knapsacks" are of bigger capacities, *i.e.*, much longer, the embedding optimization in the next phase by the MKP will be better. Staring from this point, we develop two decomposition methods to obtain longer decomposed paths for the O2O and M2O embeddings respectively.

The O2O embedding

We extract a substrate path P^s by finding the longest path in a Depth-first Search Tree (DST) of G^s. We repeat the process by keeping extracting substrate paths until G^s is completely decomposed into a set of P_i^s, $1 \leq i \leq m$, where m is the number of substrate paths obtained.

The M2O embedding

For the M2O embedding, we can utilize the substrate trails as the substrate "path" as proven above. Therefore, we try to obtain the substrate trail as long as possible. Since Euler trail is the longest one, we first add $\frac{o(G^s)}{2} - 1$ new edges to connect $\frac{o(G^s)}{2} - 1$ pairs of SNs of odd degree, then utilize an Eulerian trail algorithm to expand it. Finally, after deleting these newly added edges, we decompose G^s into a set of trails T_i^s, $1 \leq i \leq m$, where m is the number of trails obtained.

4.5.2 Embedding by MKP

After the path decomposition phase, we regard each path VNR P_j^r as an item with size $|P_j^r|$ and profit w_j, and treat each substrate path P_i^s or trail T_i^s as a knapsack with capacity $|P_i^s|$ or $|T_i^s|$ in the O2O or M2O embeddings respectively as shown in Fig. 4.12. Obviously, it is an m-knapsacks-n-items MKP, where $m \leq |E^s|$.

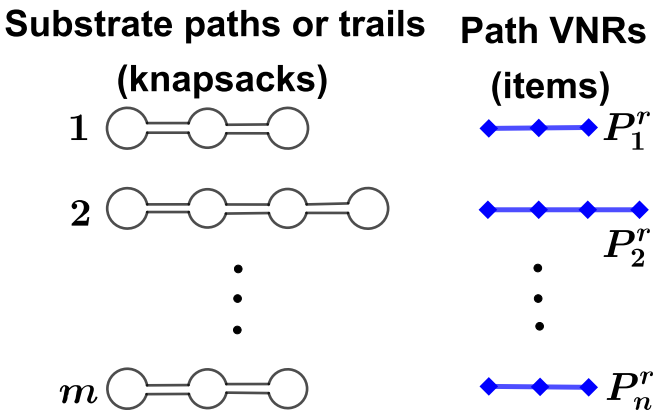


Figure 4.12: *MKP embedding.*

4.5.3 Resource Assignment by MDKP

After embedding some path VNRs by MKP without considering CPUs and BWs, we need to assign the corresponding demanded CPU and BW resources to as many embedded path VNRs as possible. Here, we treat each embedded P_j^r as an item with $(|V^s| + |E^s|)$-dimensional size attributes: $(s_{jv_1^s}, s_{jv_2^s}, ..., s_{jv_{|V^s|}^s}, s_{je_1^s}, s_{je_2^s}, ..., s_{je_{|E^s|}^s})$. For the first $|V^s|$ attributes, if some VNs v^r of the P_j^r are embedded on an SN, say v_k^s, then the attribute $s_{jv_k^s} = \sum_{v^r \to v_k^s} CPU(v^r)$, otherwise 0. For the last $|E^s|$ attributes, if a VL e^r of the P_j^r is embedded on an SL, say e_l^s, then the attribute $s_{je_l^s} = BW(e^r)$, otherwise 0. Finally, the array of capacity attributes of the knapsack is that: $(CPU(v_1^s), ..., CPU(v_{|V^s|}^s), BW(e_1^s), ..., BW(e_{|E^s|}^s))$. How to assign resources to these embedded path VNRs to maximize revenue is obviously a $(|V^s| + |E^s|)$-DKP with n_1 items as shown in Fig. 4.13, where $n_1 \leq n$ is the number of embedded path VNRs by MKP.

4.5.4 Final Assembled Algorithm and Time Complexity

After the resource assignment, we can update the CPU and BW of each SN and SL, resulting in a remained substrate network. We then continue the whole process, from path decomposition to resource assignment, to embed the rest path VNRs until no more paths can be embedded. The final assembled algorithm is shown in Algorithm 4.1.

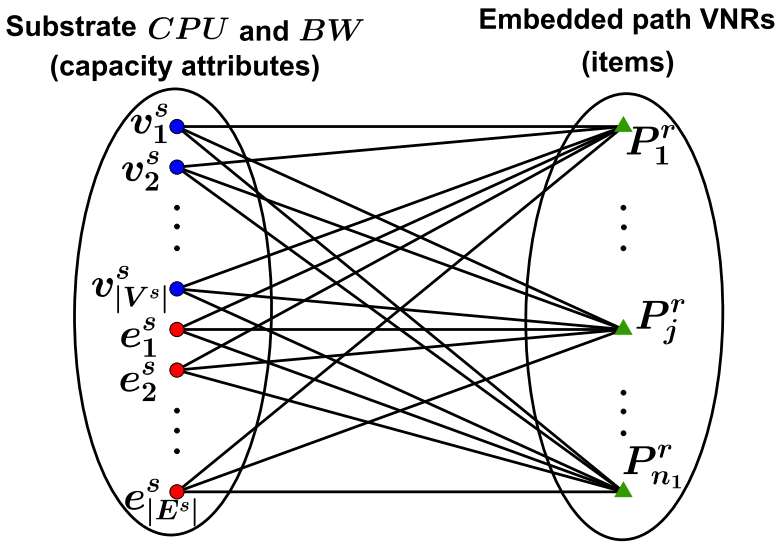


Figure 4.13: *MDKP assignment.*

Algorithm 4.1: Procedure of Path-Embedding (PE)

Input : A substrate network $G^s(V^s, E^s)$ and a set of path VNRs $\{P_1^r, P_2^r, ..., P_n^r\}$.
Output: Final revenue.

```
set Flag ← ture;
while Flag do
    run Path Decomposition Phase on G^s;
    run Embedding by MKP;
    run Resource Assignment by MDKP;
    if no path VNR can be embedded then
        Flag ← false;
    end
    update the substrate network G^s;
end
```

The time complexity of the path decomposition phase is as follows:

- The *O2O* embedding: The time complexity of extracting one substrate path by constructing a *DST* is $\mathcal{O}(|V^s| + |E^s|)$ and there are at most $|E^s|$ paths extracted. Hence, the total time complexity is $\mathcal{O}(|E^s| \times (|V^s| + |E^s|))$.
- The *M2O* embedding: The time complexity is combined by connecting pairs of SNs of odd-degree and expanding the Eulerian trail, that is $\mathcal{O}(|E^s|)$ [13]. Therefore, the overall time complexity is $\mathcal{O}(|V^s| + |E^s|)$.

The time complexity of embedding by MKP and resource assignment by MDKP, depending on the algorithms for solving MKP and MDKP, are bounded by $T_{MKP}(|E^s|, n)$ and $T_{MDKP}(|V^s| + |E^s|, n)$ respectively [42], where n is the number of path VNRs. Net-

work operators can select algorithms according to their computing capability. ([42] elaborates most of the current algorithms for MKP and MDKP.)

The time complexity for updating the substrate network is $\mathcal{O}(|V^s| + |E^s|)$, and we repeat the **while**-loop at most n times. Thus, the time complexity of Procedure PE is $\mathcal{O}\big(n \times (|E^s|^2 + T_{MKP}(|E^s|, n) + T_{MDKP}(|V^s| + |E^s|, n)\big)$ for the *O2O* embedding and $\mathcal{O}\big(n \times (|E^s| + T_{MKP}(|E^s|, n) + T_{MDKP}(|V^s| + |E^s|, n)\big)$ for the *M2O* embedding respectively. In this chapter, we use PEM2O and PEO2O to denote the path embedding algorithms for the M2O and O2O embeddings respectively.

4.6 Cycle Embedding

In this section, we investigate the cycle embedding problem when the substrate network is also a cycle, *i.e.*, $Emb(C^s, \{C^r_1, C^r_2, ..., C^r_n\})$ where C^s is the substrate cycle and $\{C^r_1, C^r_2, ..., C^r_n\}$ is the set of cycle VNRs. (*CPU*s and *BW*s are arbitrary.) In this chapter, we focus on a natural embedding way called *Simplex Cycle Embedding*, which evenly embeds VNs and VLs on the substrate cycle.

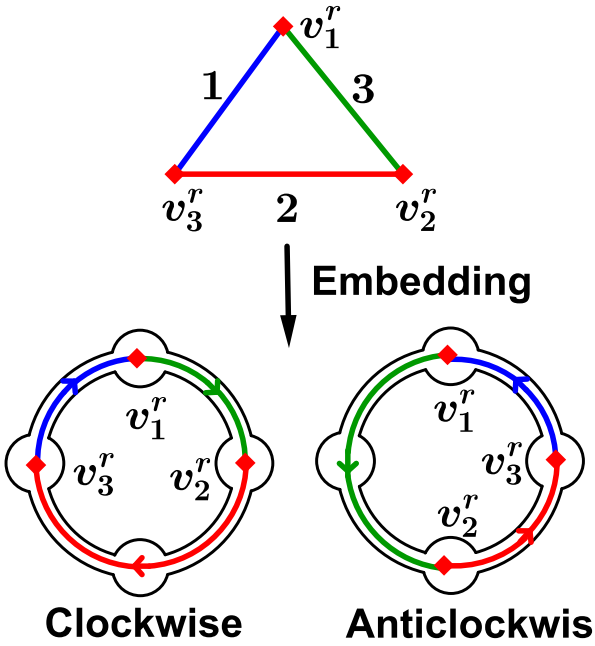


Figure 4.14: *Two simplex cycle embeddings: clockwise and anticlockwise.*

Definition 4.6.1. *Simplex Cycle Embedding: We assume the substrate cycle is $C^s(v^s_1, ..., v^s_m v^s_1)$. Given a starting SN $v^s_i \in C^s$ with one direction dir (clockwise or anticlockwise denoted by "+" or "-" respectively), we arrange all SNs of C^s in such a sequence, denoted by $Seq(v^s_i, dir)$, that for dir = "+", $[v^s_i, v^s_{i+1}, v^s_{i+2}, ..., v^s_{i-1}]$, and for dir = "-", $[v^s_i, v^s_{i-1}, v^s_{i-2}, ..., v^s_{i+1}]$, where all arithmetical operations of subscripts of SNs are modulo m.*

Given a cycle VNR $C^r(v^r_1, ..., v^r_n v^r_1)$, Simplex Cycle Embedding is that: $v^r_j \to v^s_{i_j}, 1 \leq j \leq$

n, which can be expressed as $\begin{pmatrix} v_1^r & ... & v_j^r & v_{j+1}^r & ... & v_n^r \\ v_{i_1}^s & ... & v_{i_j}^s & v_{i_{j+1}}^s & ... & v_{i_n}^s \end{pmatrix}$*, where the node mapping uses the O2O embedding and follows one direction dir. More specifically,* $\forall j, v_{i_j}^s$ *is ahead of* $v_{i_{j+1}}^s$ *in* $Seq(v_{i_1}^s, dir)$*. Figure 4.14 illustrates a triangle VNR embedded on a substrate cycle in clockwise and anticlockwise simplex cycle embeddings, where the numbers beside VLs indicate the demanded BWs.*

Similarly, given a C^s and a C^r, $Emb(C^s, C^r)$ is the elementary and primary problem in simplex cycle embedding. The similar problem $Emb(G^s, P^r)$ of path embedding is $\mathcal{NP}$-hard even in the preliminary model. Is $Emb(C^s, C^r)$ also $\mathcal{NP}$-hard? Moreover, even the answer of $Emb(C^s, C^r)$ is **Yes**, different embedding ways could result in different resource consumptions: In Fig. 4.14, the BW consumption of the clockwise is $1 + 2 \times 2 + 3 = 8$ while that for the anticlockwise is $2 \times 3 + 2 + 1 = 9$. Can we efficiently find the least-resource-consuming embedding way? How about the AcR and Rev problems in cycle-to-cycle embedding?

Following the three important questions, we unfold this section as follows.

- First, we construct a Weighted Directed Auxiliary Graph (WDAG) in polynomial-time and prove that each of its directed cycles corresponds to a feasible simplex cycle embedding.
- Then, the minimum weighted directed cycle corresponds to the least-resource-consuming embedding, which can be obtained by dynamic programming in polynomial time.
- Finally, we prove that both the AcR and Rev problems are strongly $\mathcal{NP}$-hard, and thus devise effective heuristic algorithms to solve them.

Given a substrate cycle $C^s(v_1^s, ..., v_m^s v_1^s)$ and a cycle VNR $C^r(v_1^r, ..., v_n^r v_1^r)$, for all v_j^r, let $\mathcal{F}_{v_j^r} = \{v_i^s \in V^s | CPU(v_i^s) \geq CPU(v_j^r)\}$, *i.e.*, the set of feasible SNs on which v_j^r can be embedded, and for all $v_j^r v_{j+1}^r$, $\mathcal{F}_{v_j^r v_{j+1}^r} = \{e^s \in E^s | BW(e^s) \geq BW(v_j^r v_{j+1}^r)\}$, *i.e.*, the set of feasible SLs whose BW is not smaller than $v_j^r v_{j+1}^r$'s. If $Emb(C^s, C^r)$ is **Yes** in simplex cycle embedding, there must exist an embedding way that $v_1^r \to v_{i_1}^s \in \mathcal{F}_{v_1^r}$, following one direction *dir*. With respect to the condition that $v_1^r \to v_{i_1}^s \in \mathcal{F}_{v_1^r}$ and the embedding direction *dir*, we construct a WDAG denoted by $\hat{G}_{dir}^{v_{i_1}^s}(\hat{V}, \hat{A})$, where $\hat{V}$ is the vertex set and $\hat{A}$ is the arc set.

1) The vertex set $\hat{V}$ comprises of n parts $\{\hat{\mathcal{F}}_{v_j^r} | _{j=1}^n\}$, where the j-th part $\hat{\mathcal{F}}_{v_j^r}$ corresponds to the set $\mathcal{F}_{v_j^r}$. Except $\hat{\mathcal{F}}_{v_1^r}$, there is a one-to-one mapping, denoted by MP, between vertices in $\hat{\mathcal{F}}_{v_j^r}$ and SNs in $\mathcal{F}_{v_j^r}$. In other words, $\forall \hat{v} \in \hat{\mathcal{F}}_{v_j^r}$ one-to-one corresponds to $MP(\hat{v}) \in \mathcal{F}_{v_j^r}$. In $\hat{\mathcal{F}}_{v_1^r}$, there is only one vertex, $\hat{v}_1$, which corresponds to $v_{i_1}^s$, *i.e.*, $MP(\hat{v}_1) = v_{i_1}^s$ as shown in Fig. 4.15.

2) The arc set $\hat{A}$ is iteratively constructed as below: First starting at $\hat{v}_1$, for each such vertex in $\hat{\mathcal{F}}_{v_2^r}$, say $\hat{v}_2$, that satisfies two criteria with $\hat{v}_1$, an arc is constructed with $\hat{v}_1$ as tail and $\hat{v}_2$ as head. The two criteria are as follows. **Criterion 1:** $MP(\hat{v}_1)$ is ahead of

$MP(\hat{v}_2)$ in $Seq(v^s_{i_1}, dir)$. **Criterion 2:** each SL e^s, lying in the segment from $MP(\hat{v}_1)$ to $MP(\hat{v}_2)$ following dir, belongs to $\mathcal{F}_{v^r_1 v^r_2}$. Besides, a weight is assigned to this arc which equals $|MP(\hat{v}_1) - MP(\hat{v}_2)| \times BW(v^r_1 v^r_2)$, where $|MP(\hat{v}_1) - MP(\hat{v}_2)|$ is the number of SLs in the segment from $MP(\hat{v}_1)$ to $MP(\hat{v}_2)$ following dir. Next, for each such vertex in $\hat{\mathcal{F}}_{v^r_2}$ with incoming edges, say $\hat{v}_2$ and $d^-(\hat{v}_2) > 0$, we repeat the same procedure on it as we did for $\hat{v}_1$: Searching those vertices in $\hat{\mathcal{F}}_{v^r_3}$ which satisfy the two criteria with $\hat{v}_2$; and arcs are constructed with $\hat{v}_2$ as tail; and weights are computed and assigned to these arcs. After some iterations, at certain vertex part, say $\hat{\mathcal{F}}_{v^r_j}, j < n$, if there is no vertex in $\hat{\mathcal{F}}_{v^r_j}$ whose indegree is greater than 0, then the whole process is terminated. Otherwise, we reach the vertex part $\hat{\mathcal{F}}_{v^r_n}$. For each such vertex in $\hat{\mathcal{F}}_{v^r_n}$ with non-zero indegree, say $\hat{v}_n$ and $d^-(\hat{v}_n) > 0$, an arc with the corresponding weight is constructed with $\hat{v}_n$ as tail and $\hat{v}_1$ as head, if $\hat{v}_n$ satisfies **Criterion 2** with $\hat{v}_1$, *i.e.*, each SL e^s lying in the segment from $MP(\hat{v}_n)$ to $MP(\hat{v}_1)$ following dir, belongs to $\mathcal{F}_{v^r_n v^r_1}$. Figure 4.15 shows a complete WDAG $\hat{G}^{v^s_{i_1}}_{dir}$, and Algorithm 4.2 formally gives the construction of WDAG $\hat{G}^{v^s_{i_1}}_{dir}$.

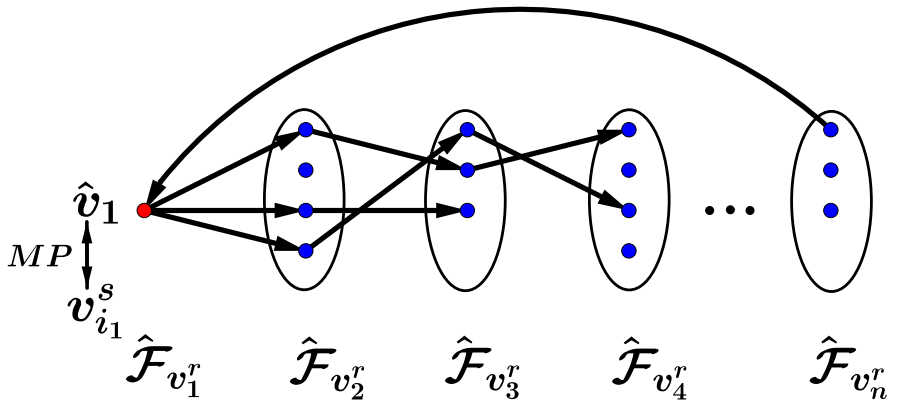


Figure 4.15: *The WDAG $\hat{G}^{v^s_{i_1}}_{dir}$ with respect to $v^s_{i_1}$ and dir.*

Theorem 4.6.1. *The time complexity of Algorithm 4.2 is $\mathcal{O}(m^3 n)$, where m and n are the SN and the VN numbers of the substrate cycle C^s and the cycle VNR C^r respectively.*

Proof. The time complexity of Algorithm 4.2 consists of two parts: 1) establishing the vertex set $\hat{V}$; 2) constructing the weight arc set $\hat{A}$. For the first part, the *Lines 1-2* set up the n vertex parts $\hat{\mathcal{F}}_{v^r_j}, \forall 1 \leq j \leq n$, and build a one-to-one mapping MP between vertices in $\hat{\mathcal{F}}_{v^r_j}$ and $\mathcal{F}_{v^r_j}$. Since each $|\hat{\mathcal{F}}_{v^r_j}| \leq m$, time consumption of this part is up to $\mathcal{O}(mn)$. For the second part, in the worst case, for each vertex pair $(\hat{v}_j, \hat{v}_{j+1})$ between $\hat{\mathcal{F}}_{v^r_j}$ and $\hat{\mathcal{F}}_{v^r_{j+1}}$, we need to check **Criterion 1** and **Criterion 2** to decide whether to construct an arc. The time consumption of checking **Criterion 1** and **Criterion 2** is $\mathcal{O}(m)$. There are at most $m^2 n$ arcs, and thus the time consumption of constructing the arc set $\hat{A}$ is $\mathcal{O}(m^3 n)$. Combing $\mathcal{O}(mn)$ and $\mathcal{O}(m^3 n)$, the total time complexity is $\mathcal{O}(m^3 n)$. □

Algorithm 4.2: Procedure of Constructing the Weighted Directed Auxiliary Graph (C-WDAG)

Input : C^s, C^r, and a vertex $v_{i_1}^s \in \mathcal{F}_{v_1^r}$ as well as a direction dir.

Output: The $\hat{G}_{dir}^{v_{i_1}^s}$.

set $\hat{\mathcal{F}}_{v_1^r} \leftarrow \hat{v}_1$ and $MP(\hat{v}_1) \leftarrow v_{i_1}^s$;
set $\hat{\mathcal{F}}_{v_j^r}$ and $MP(\hat{\mathcal{F}}_{v_j^r}) \leftarrow \mathcal{F}_{v_j^r}$ $\forall 2 \leq j \leq n$;
set $Flag \leftarrow$ **true**, $j \leftarrow 1$;
while $Flag$ **do**
  if $j = 1$ **then**
    set $\mathcal{V}_2 \leftarrow \{\hat{v}_2 | \hat{v}_2 \in \hat{\mathcal{F}}_{v_2^r}$ & satisfying **Criterion 1** and **Criterion 2** with $\hat{v}_1\}$;
    for $\hat{v}_2 \in \mathcal{V}_2$ **do**
      set an arc with $\hat{v}_1$ as tail and $\hat{v}_2$ as head;
      assign arc weight $|MP(\hat{v}_1) - MP(\hat{v}_2)| \times BW(v_1^r v_2^r)$;
    end
    $j \leftarrow j + 1$;
  end
  else if $2 \leq j \leq n - 1$ **then**
    set $\mathcal{V}_j \leftarrow \{\hat{v}_j | \hat{v}_j \in \hat{\mathcal{F}}_{v_j^r}$ & $d^-(\hat{v}_j) > 0\}$;
    if $\mathcal{V}_j = \emptyset$ **then**
      $Flag \leftarrow$ **false**;
    end
    else
      for $\hat{v}_j \in \mathcal{V}_j$ **do**
        set $\mathcal{V}_{j+1} \leftarrow \{\hat{v}_{j+1} | \hat{v}_{j+1} \in \hat{\mathcal{F}}_{v_{j+1}^r}$ & satisfying **Criterion 1** and **Criterion 2** with $\hat{v}_j\}$;
        for $\hat{v}_{j+1} \in \mathcal{V}_{j+1}$ **do**
          set an arc with $\hat{v}_j$ as tail and $\hat{v}_{j+1}$ as head;
          assign arc weight $|MP(\hat{v}_j) - MP(\hat{v}_{j+1})| \times BW(v_j^r v_{j+1}^r)$;
        end
      end
      $j \leftarrow j + 1$;
    end
  end
  else if $j = n$ **then**
    set $\mathcal{V}_n \leftarrow \{\hat{v}_n | \hat{v}_n \in \hat{\mathcal{F}}_{v_n^r}$ & $d^-(\hat{v}_n) > 0$ & satisfying **Criterion 2** with $\hat{v}_1\}$;
    for $\hat{v}_n \in \mathcal{V}_n$ **do**
      set an arc with $\hat{v}_n$ as tail and $\hat{v}_1$ as head;
      assign arc weight $|MP(\hat{v}_n) - MP(\hat{v}_1)| \times BW(v_n^r v_1^r)$;
    end
    $Flag \leftarrow$ **false**;
  end
end
return $\hat{G}_{dir}^{v_{i_1}^s}$.

The WDAG $\hat{G}_{dir}^{v_{i_1}^s}$ constructed in Algorithm 4.2 has an important property described in Theorem 4.6.2.

Theorem 4.6.2. *Given a substrate cycle $C^s(v_1^s, ..., v_m^s v_1^s)$ and a cycle VNR $C^r(v_1^r, ..., v_n^r v_1^r)$, under the condition that $v_1^r \rightarrow v_{i_1}^s \in \mathcal{F}_{v_1^r}$ and following dir, there is a one-to-one relation between each directed cycle in the WDAG $\hat{G}_{dir}^{v_{i_1}^s}$ and each feasible simplex cycle embedding way.*

Proof. For each feasible simplex cycle embedding way under the condition that $v_1^r \rightarrow v_{i_1}^s$ and following *dir*, *w.l.o.g.*, we assume it is $\begin{pmatrix} v_1^r & v_2^r & ... & v_j^r & v_{j+1}^r & ... & v_n^r \\ v_{i_1}^s & v_{i_2}^s & ... & v_{i_j}^s & v_{i_j+1}^s & ... & v_{i_n}^s \end{pmatrix}$, where each $v_j^r \rightarrow v_{i_j}^s$ and $v_{i_j}^s \in \mathcal{F}_{v_j^r}$. Since MP is a one-to-one mapping from $\hat{\mathcal{F}}_{v_j^r}$ to $\mathcal{F}_{v_j^r}$, we use MP^{-1} to represent the inverse, *i.e.*, $MP^{-1}(v_{i_j}^s) \in \hat{\mathcal{F}}_{v_j^r}, MP(MP^{-1}(v_{i_j}^s)) = v_{i_j}^s$. As it is feasible, $MP^{-1}(v_{i_{j+1}}^s)$ must satisfy **Criterion 1** and **Criterion 2** with $MP^{-1}(v_{i_j}^s)$. Therefore, according to the construction process, there is an arc with $MP^{-1}(v_{i_j}^s)$ as tail and $MP^{-1}(v_{i_{j+1}}^s)$ as head, and $\left(MP^{-1}(v_{i_1}^s), ..., MP^{-1}(v_{i_j}^s), ..., MP^{-1}(v_{i_n}^s) MP^{-1}(v_{i_1}^s)\right)$ forms a directed cycle in $\hat{G}_{dir}^{v_{i_1}^s}$.

For each directed cycle in $\hat{G}_{dir}^{v_{i_1}^s}$, say $\hat{C}$, according to the construction process, $\hat{C}$ must pass through exactly one vertex in each vertex part $\hat{F}_{v_j^r}, \forall 1 \leq j \leq n$, say $(\hat{v}_1, ..., \hat{v}_j, ..., \hat{v}_n)$. Since $\hat{v}_{j+1}$ satisfies **Criterion 1** and **Criterion 2** with $\hat{v}_j$, it is obvious that $MP(\hat{C}) = \begin{pmatrix} v_1^r & ..... & v_j^r & ..... & v_n^r \\ MP(\hat{v}_1) ... MP(\hat{v}_j) ... MP(\hat{v}_n) \end{pmatrix}$ must be a feasible simplex cycle embedding way. □

Subsequently, combining with the weights assigned to arcs of the WDAG $\hat{G}_{dir}^{v_{i_1}^s}$, the sum of arc weights of a directed cycle in $\hat{G}_{dir}^{v_{i_1}^s}$ is equal to the total *BW* consumption of the embedding way which corresponds to the directed cycle. Since the *CPU* consumption is fixed to the sum of all demanded *CPUs* of VNs, to obtain the least-resource-consuming embedding way, we just need to search the minimum weighted directed cycle in the WDAG $\hat{G}_{dir}^{v_{i_1}^s}$ which can be solved by dynamic programming in polynomial time.

Moreover, to obtain the optimal least-resource-consuming simplex cycle embedding, we just need to construct $2 \times |\mathcal{F}_{v_1^r}|$ WDAGs (two directions), search the directed cycle with the minimum weigth in each WDAG, and finally output the smallest one among them. We formally give Algorithm 4.3.

In *Lines 1-2*, for each VN v_j^r and VL $v_j^r v_{j+1}^r$, we set up the feasible SN sets and SL sets. The time complexity of *Lines 1-2* is $\mathcal{O}(mn)$. At *Lines 3*, we set a variable *EMB* to record the optimal embedding way whose initial value is $\emptyset$ and another variable *Cost* to record the *BW* consumption of the *EMB* whose initial value is large enough denoted by ∞. In *Lines 4-9*, for each $v_{i_1}^s \in \mathcal{F}_{v_1^r}$ and each direction *dir* (**"+"** or **"-"**), we construct the corresponding WDAG by procedure C-WDAG at *Line 6* whose time complexity is $\mathcal{O}(m^3 n)$ by Theorem 4.6.1. At *Line 7*, we search the minimum weighted directed

Algorithm 4.3: Procedure of Cycle-to-Cycle Embedding (C2CE)

Input : $C^s(v_1^s, ..., v_m^s v_1^s)$ and $C^r(v_1^r, ..., v_n^r v_1^r)$
Output: The least-resource-consuming embedding way.

1 **set** $\mathcal{F}_{v_j^r} \leftarrow \{v_i^s | CPU(v_i^s) \geq CPU(v_j^r)\}, \forall j$;
2 **set** $\mathcal{F}_{v_j^r v_{j+1}^r} \leftarrow \{v_i^s v_{i+1}^s | BW(v_i^s v_{i+1}^s) \geq BW(v_j^r v_{j+1}^r)\}$;
3 **set** $EMB \leftarrow \emptyset$, $Cost \leftarrow \infty$;
4 **for** $v_{i_1}^r \in \mathcal{F}_{v_1^r}$ **do**
5 **for** *dir ("+" or "-")* **do**
6 **construct** the WDAG by Procedure C-WDAG;
7 **search** the minimum weighted directed cycle $\hat{C}$ in the WDAG;
8 **if** $w(\hat{C}) < Cost$ **then**
9 $EMB \leftarrow MP(\hat{C})$, $Cost \leftarrow w(\hat{C})$;
10 **end**
11 **end**
12 **end**
13 **return** EMB;

cycle $\hat{C}$, and its time complexity is $\mathcal{O}(m^2 n)$ by dynamic programing. At *Lines 8-9*, if the weight of $\hat{C}$ denoted by $w(\hat{C})$ is less than current $Cost$, we replace $Cost$ by $w(\hat{C})$ and EMB by the embedding way denoted by $MP(\hat{C})$ which corresponds to $\hat{C}$. Finally, at *Line 10*, we return the EMB (if $EMB = \emptyset$ then C^r can not be embedded on C^s by simplex cycle embedding). The total time complexity of Algorithm 4.3 is $\mathcal{O}(mn) + 2m \times \left(\mathcal{O}(m^3 n) + \mathcal{O}(m^2 n)\right) + \mathcal{O}(1) = \mathcal{O}(m^4 n)$, where m and n are the SN and the VN numbers of the substrate cycle C^s and the cycle VNR C^r respectively.

Notice that the WDAG can be applied in simplex cycle embedding in the S-VNE by slightly modifying $\mathcal{F}_{v^r} = \{v^s \in \Phi(v^r) | CPU(v^s) \geq CPU(v^r)\}$. Thus $Emb(C^s, C^r)$ in the S-VNE can be also solved in polynomial time.

For the $Emb(C^s, C^r)$ problem in simplex cycle embedding, we can solve it in polynomial time. How about the AcR and Rev problem? Unfortunately, both of them are still $\mathcal{NP}$-hard.

Theorem 4.6.3. *In cycle-to-cycle embedding, both the AcR and Rev problems are $\mathcal{NP}$-hard. Moreover, the hardnesses of each problem is no less than any d-DKP, where d is any constant integer.*

Proof. Since the AcR problem is a special case of the Rev problem, we just need to prove the AcR problem is $\mathcal{NP}$-hard. To this end, we polynomial-timely reduce the $\mathcal{NP}$-hard problem "Cardinality d-DKP" [42] to the AcR problem. Cardinality d-DKP is a special d dimensional MDKP, *i.e.*, the knapsack is with a d-dimensional capacity attributes (b_1,...,b_i,...,b_d) and each j-th item is with a d-dimensional size attributes $(s_{j1}, ..., s_{ji}, ..., s_{jd})$. The objective is to maximize the number of packed items.

Given an instance in Cardinality d-DKP, we construct the substrate cycle in such way: There are d SNs in the substrate cycle $C^s(v_1^s, ..., v_i^s, ..., v_d^s)$, and $CPU(v_1^s) = b_1$ and

$CPU(v_i^s) = B_i \times b_i, \forall 1 < i \leq d$ where B_i are relatively big numbers explained later. Each SL is with $BW = n$, *i.e.*, the number of items. We construct n cycle VNRs in such way: For the j-th cycle VNR $C^r(v_{j1}^r, ..., v_{ji}^r, ..., v_{jd}^r)$, there are d VNs, and $CPU(v_{j1}^r) = s_{j1}$ and $CPU(v_{ji}^r) = B_i \times s_{ji}, \forall 1 < i \leq d$ such that $B_i \times s_{ji} > \max_{1<k<i}(B_k \times b_k, b_1)$ (by setting $CPUs$ like this the v_{ji}^r can only be embedded on v_i^s). Each VL is with $BW = 1$. Thus, the solution of the instance of Cardinality d-DKP is equivalent to that of the AcR problem. □

As shown in [42], even the 2-DKP is strongly $\mathcal{NP}$-hard and the hardness of solving d-DKP keeps entrenched with the increase of d. To effectively solve the strongly $\mathcal{NP}$-hard problem, we herein develop a heuristic algorithm based on the optimization for single cycle embedding as follows. Intuitively, for a cycle C^r, if its ratio of revenue to resource consumption is higher than the others, it tends to be embedded so as to achieve a more efficient income for the InP. This consists of the main motivation of our greedy strategy in Algorithm 4.4: Given a substrate cycle C^s and a set of cycle VNRs $\{C_1^r, C_2^r, ..., C_n^r\}$, for each C_j^r, we first estimate the ratio of revenue to resource consumption, *i.e.*, $\frac{w_j}{\sum_{v^r \in C_j^r} CPU(v^r) + \sum_{e^r \in C_j^r} BW(e^r)}$. We then arrange them in the descending order of the estimated ratios, and sequentially embed them on the C^s by procedure C2CE until no more cycle VNR can be embedded by simplex cycle embedding. However, one thing should be noted that the simplex cycle embedding has its own shortage, *i.e.*, it limits the solution space. Therefore, if no cycle VNR can be embedded by simplex cycle embedding, we continue the embedding by running general algorithms. Via this combination, both merits of simplex cycle embedding and general algorithms can be conflated.

Algorithm 4.4: Procedure of Greedy Revenue (GR)

Input : C^s and $\{C_1^r, C_2^r, ..., C_n^r\}$.
Output: The final revenue.
1 **set** cycle VNRs in descending order by estimated ratios;
2 **run** C2CE to sequentially embed cycle VNRs until can't;
3 **run** general algorithms;
4 **return** final revenue;

4.7 Numerical Results

In this section, we compare our proposed algorithms PEO2O and PEM2O respectively for path embedding and GR for cycle embedding to the existing general algorithms. Two general embedding algorithms from [17] and [77], denoted by RW and BA respectively, are used as our benchmarks. Here, we use GRRW and GRBA to denote the procedures GR which invoke RW and BA respectively. We repeat each simulation 50 times under the same circumstance to ensure sufficient statistical accuracy, and a 95%

confidence interval is given to each numerical result. All the simulations have been run by MATLAB 2015a on a computer with 3.2 GHz Intel(R) Core(TM) i5-4690S CPU and 8 GBytes RAM.

4.7.1 Evaluation Environments

Path Embedding

Substrate Networks We use the GT-ITM tool [79], prevailing in the generation of practical network topologies, to randomly generate two substrate networks denoted by G_1^s, G_2^s respectively. Both of these substrate networks have 100 SNs and 1000 SLs, corresponding to a medium-sized ISP. Besides, we also use a complete graph of 100 SNs as the substrate network denoted by CG^s. The CPU and BW of each SN and SL are set as 100 units.

Virtual Network Requests The length of each path VNR is randomly generated in the range of $[5, 10]$. The CPU and BW of each VN and VL are randomly generated in the range of $[1, 5]$ units. The number of path VNRs is set as 1000 in each simulation.

Cycle Embedding

Substrate Networks We set up three substrate cycles denoted by C_{20}^s, C_{25}^s, C_{30}^s respectively, whose number of SNs are 20, 25 and 30 respectively, corresponding to the sizes of existing substrate optical rings. The CPU and BW of each SN and SL are set as 100 units.

Virtual Network Requests The number of VNs of each cycle VNR is randomly generated in the range of $[5, 10]$. The CPU and BW of each VN and VL are randomly generated in the range of $[1, 5]$ units. The number of cycle VNRs is set as 100 in each simulation.

Performance Metrics

The comparisons are performed for both the AcR and Rev problems.

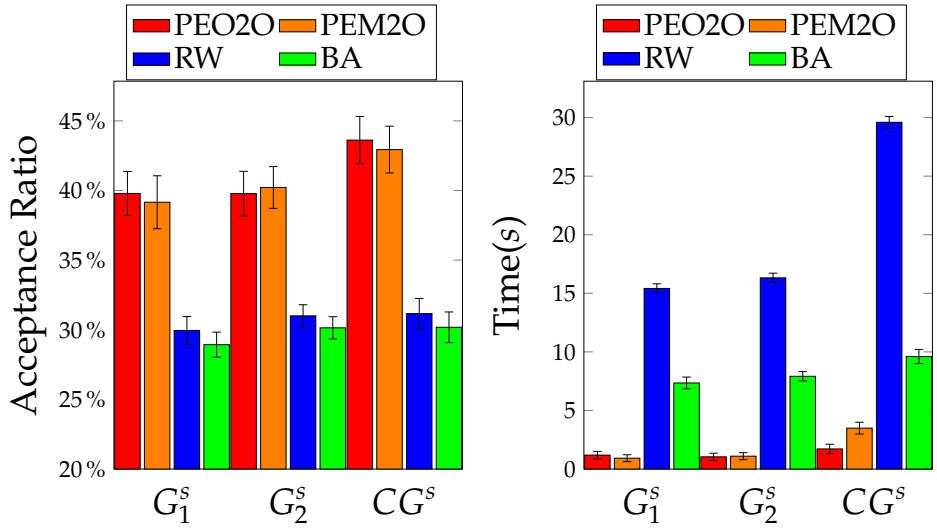
- The AcR problem: The revenue of each VNR is set to be one. Besides, we tweak the objective function of the AcR problem as $\frac{|S|}{n}$, where S is the subset of embedded VNRs and n is the number of total VNRs.

- The Rev problem: The revenue of each VNR is proportional to its VN number in the range of [5,10].

4.7.2 Simulation Results

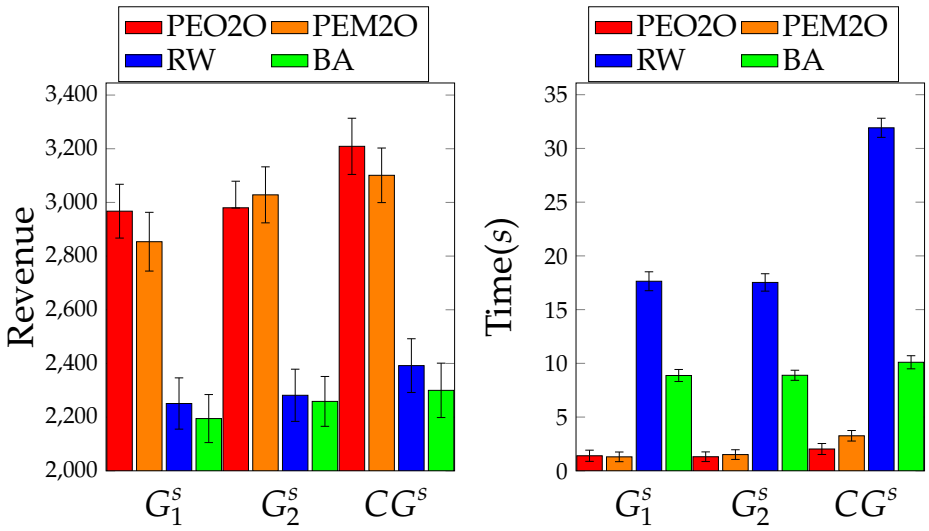
Path Embedding

Figures 4.16 and 4.17 respectively demonstrate the numerical results of the AcR and Rev problems in path embedding. The average acceptance ratio of all the three substrate networks, as shown in Fig 4.16(a), are 41.06% and 40.77% for PEO2O and PEM2O respectively, compared to 30.70% and 29.75% for RW and BA respectively. The average revenue, as shown in Fig. 4.17(a), are 3052.12 and 2994.28 for PEO2O and PEM2O respectively, compared to 2308.05 and 2251.01 for RW and BA respectively. For time complexity as shown in Figs. 4.16(b) and 4.17(b), both PEO2O and PEM2O are comparable with an average run time of 1.32s and 1.84s respectively, obviously smaller than that of RW and BA, 20.44s and 8.28s. We can also find that PEO2O outperforms PEM2O in G_1^s, while it is inverse in G_2^s, which indicates that their performance depends on the topologies of the substrate networks. In summary, PEO2O and PEM2O are much more efficient and effective than the two general algorithms RW and BA in path embedding.



(a) Acceptance ratio (b) Time complexity

Figure 4.16: *Numerical results of the AcR problem in path embedding.*

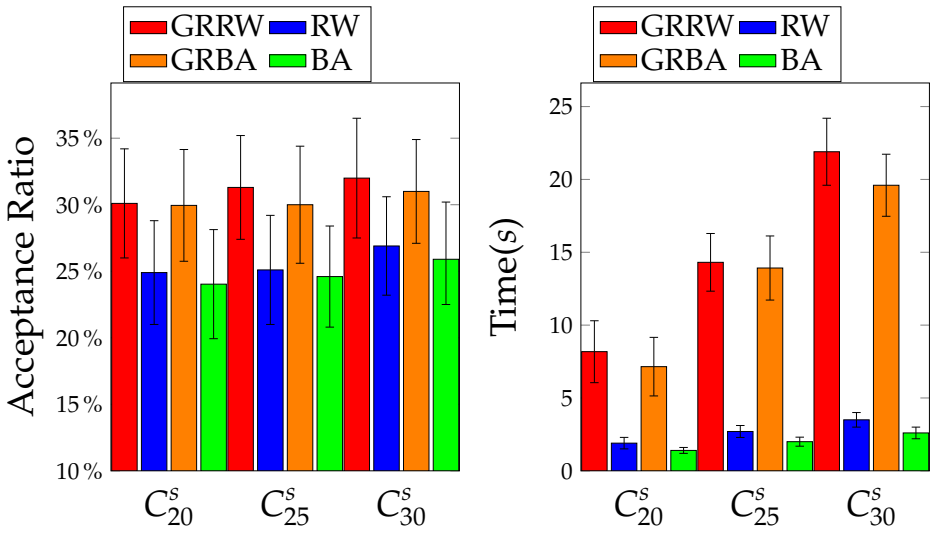


(a) Revenue (b) Time complexity

Figure 4.17: *Numerical results of the Rev problem in path embedding.*

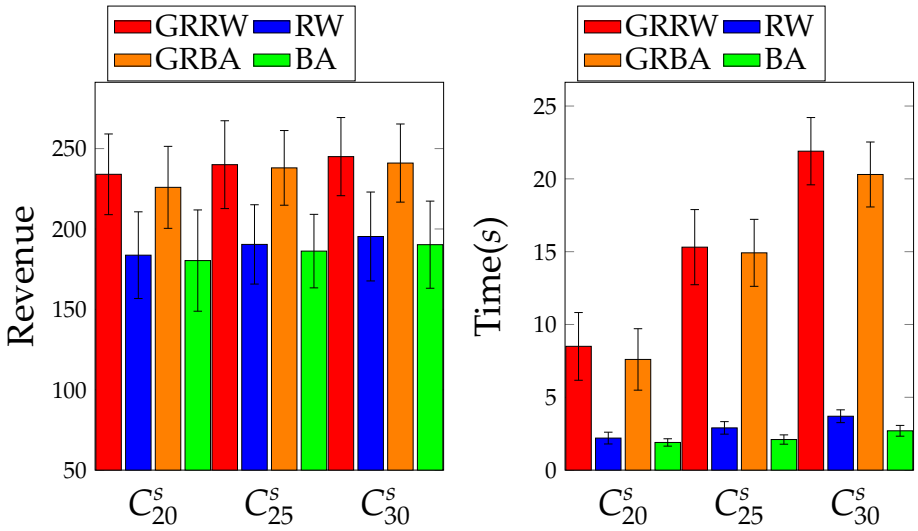
Cycle Embedding

Figures 4.18 and 4.19 respectively showcase the numerical results of the AcR and Rev problems in cycle embedding. The average acceptance ratio, as shown in Fig. 4.18(a), of GRRW and GRBA are 31.13% and 30.31% respectively compared to 25.63% and 24.84% of RW and BA respectively. The average revenue, as shown in Fig. 4.19(a), of GRRW and GRBA are 239.68 and 234.96 respectively while 189.80 and 185.62 of RW and BA respectively. From the aspect of final results of acceptance ratios and revenues, GRRW and GRBA take advantage over RW and BA. For the time complexity as shown in Figs. 4.18(b) and 4.19(b), while the run times of RW and BA are relatively stable and smaller than 2.5s, that of GRRW and GRBA are quickly climbing because the time complexity of construction of the WDAG is fourth-order about the number of SNs. But, the corresponding acceptance ratios and revenues do not improve much with the increase of run times of GRRW and GRBA. Thus, in general substrate networks, in the future more work is needed to balance the size of decomposed substrate cycles and develop decomposition strategies so as to constitute cycle embedding algorithms as competitive as PEO2O and PEM2O for path embedding.



(a) Acceptance ratio (b) Time complexity

Figure 4.18: *Numerical results of the AcR problem in cycle embedding.*



(a) Revenue (b) Time complexity

Figure 4.19: *Numerical results of the Rev problem in cycle embedding.*

4.8 Conclusions

In this chapter, we systematically investigated the VNE problems in path and cycle topologies. For path embedding, we proved its $\mathcal{NP}$-hardnesses for both the O2O and M2O embeddings. Leveraging Eulerian trail, we developed some approximation algo-

rithms for the AcR and Rev problems in the preliminary model. In the realistic settings, we proved the inapproximability of path embedding in Theorem 4.5.1. Following the idea of expanding substrate networks into "paths", we further developed the MKP-MDKP-based algorithms for the path embedding, which turn out to be more efficient and effective than its counterparts. Regarding cycle embedding, we proposed an auxiliary graph WDAG, based on which we are able to characterize the one-to-one relation between a directed cycle in WDAG and a feasible simplex cycle embedding. Herein is devised a polynomial-time algorithm C2CE to obtain the optimal least-resource-consumption embedding solution.

Chapter 5

Conclusions and Perspectives

5.1 Summary

In this thesis, we concentrate on three challenging problems appearing in next generation communication networks. The studied problems include (1) lightpath routing and spectrum assignment for RSA in EONs and (2) the VNE for network virtualization. To solve them efficiently, we first give a theoretical analysis on them and then propose dedicated approximation algorithms based on graph theory.

In Chapter 2, we inspected how the traffic distribution and network topology impact the lightpath routing. All investigations revolve around a central concept: the conflict graph. Since whatever lightpath routing schemes are employed, the intersections among the routed lightpaths are the key matters determing the quality of optimization in the spectrum assignment. Theorem 2.3.1 bounds the optimality of the RSA by the chromatic number of the conflict graph. By Lemma 2.4.1, the first theoretical chain in Fig. 2.6 bridges the intersecting probability of lightpaths and the optimality of the RSA. Then, via the new concept conflict coefficients, the second theoretical chain in Fig. 2.8 unveils that the traffic distribution and network topology through the GOF impact the intersecting probability of lightpaths. Consequently, the two theoretical chains together figure out the synthesized impact of the two factors on the lightpath routing.

Chapter 3 investigated the DSA problem in EONs. We consider guard bands with adaptive sizes between different lightpath pairs. The main approach in this chapter is to leverage the conflict graph of the DSA problem. Through analyzing the properties of the DSA conflict graph, the $\mathcal{NP}$-hardness, inapproximability, upper and lower bounds of the DSA problem are obtained. After analyzing the theoretical characteristics of the DSA problem, we proposed the ODSA problem and transformed the DSA problem to a special case of POP. Then, a two-phased algorithm is devised to solve DSA efficiently. The first phase outputs an initial solution which is optimal in the bipartite graph while assuring a certain approximation ratio in complete conflict graph. The second phase improves the quality of the initial solution obtained in the first phase by a random approach NPM whose convergence is deducted.

In Chapter 4, we explored the topological features of the VNE under this paths and cycles. For the essential question in VNE problem $Emb(G^s, G^r)$, we proved that $Emb(G^s, P^r)$ is $\mathcal{NP}$-hard while $Emb(C^s, C^r)$ is in $\mathcal{P}$ class by the WDAG. Moreover, we give the inapproximability results about the path embedding in a preliminary model and realistic settings. In the preliminary model, unless $\mathcal{NP} = \mathcal{P}$, there is no polynomial-time algorithm of constant approximation ratio. But we proved the path embedding can be approached in terms of the vertex number of odd degree in the substrate network. In realistic settings, we proved that unless $\mathcal{NP} = \mathcal{P}$, there is no polynomial-time algorithm of approximation ratio $|E^s|$ for both the AcR and Rev problems. We further devised an efficient and effective heuristic algorithm which demonstrate obvious advantage over the general algorithms in the literature.

5.2 Future work

There are many future work which can go in-depth for these problems studied in this thesis.

For the lightpath routing, Chapter 2 just analyzes the unicast communications. The concept of conflict coefficients can also be extended to all-optical multicast communications. In general, light-trees are used to route multicast requests instead of lightpaths. The assignment of FS on light-trees can also be solved by constructing a conflict graph exactly in the same way as we discussed in the Chapter 2. Thus, similarly, we can construct a set of candidate light-trees for each multicast request, and calculate the corresponding matrix of conflict coefficients on different candidate light-trees to minimize the expected intersecting probability. But, it should be noted that the computation of candidate light-trees is more complicated than lightpaths. Besides, as analyzed in Chapter 2, the routing scheme is another key factor to influence the performance of lightpath routing. In Chapter 2, we deduced the optimal routing decision. However, this optimal routing decision is based on the assumption that the traffic distribution is obtained. In reality, how to obtain or predict the traffic distribution is another tough task which may require some other optimization techniques such as AI.

For DSA problem, as introduced in Chapter 3, it is extremely hard to approach in the general conflict graphs. But in some relatively simple graphs such as bipartite graphs, the DSA problem can be easily solved. Thus, in future work, it will be interesting to study how to properly route the lightpaths so as to obtain a simple conflict graph for the DSA problem. Besides, in this thesis, we mainly consider the min-RSA problem. Thus, its dual problem, *i.e.*, max-RSA, is also of great interest. The objective of max-RSA is to maximize the number of requests, which is also $\mathcal{NP}$-hard and requires optimization techniques based on graph theory. Furthermore, to efficiently solve the DSA problem, we proposed a two-phase algorithm, of which the second phase is a random approach. Actually, many other meta-heuristics can be utilized to improve the initial solution such as local search, simulated annealing, ant colony optimization, *etc*. To obtain a better outcome, these methods sometime can be combined together.

Through the analyses in Chapter 4, there still exist a lot of work to do for the path and cycle embeddings in the future. For example: (a) for path embedding in the realistic setting, whether there exist other approaches to improve the proposed KP-based algorithms; (b) for cycle embedding in substrate networks of general topologies, how to decompose the general substrate network G^s into a set of substrate cycles and develop efficient algorithms to embed a set of cycle VNRs $\{C_1^r, C_2^r, ..., C_n^r\}$ on a set of substrate cycles $\{C_1^s, C_2^s, ..., C_m^s\}$; (c) for the VNE problem in EONs, how to take into account the features of EONs, for instance, the continuity and contiguity of spectrum resource in optical fibers. Besides, it is of great interest to further investigate path and cycle embeddings in the S-VNE. Their theoretical hardnesses are already obtained in Chapter 4, since the G-VNE is a special case of the S-VNE. For practical algorithms, the current work for G-VNE, reflecting some topological-structure features of path and cycle embeddings, can give some inspirations to the S-VNE.

List of Figures

List of Tables

Bibliography

[1] Cisco visual networking index: Forecast and methodology, 2016–2021. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html.

[2] Itu-t recommendation g.694.1, spectral grids for wdm applications: Dwdm frequency grid, v2.0, 2012.

[3] Networkx. http://networkx.github.io/.

[4] D. Andersen. Theoretical approaches to node assignment. pages 1–13, Dec. 2002. Unpublished manuscript.

[5] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41, Apr. 2005.

[6] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. In *46th Annual IEEE FOCS*, pages 226–241, Oct. 2005.

[7] E. Archambault, N. Alloune, M. Furdek, Z. Xu, C. Tremblay, A. Muhammad, J. Chen, L. Wosinska, P. Littlewood, and M. Belanger. Routing and spectrum assignment in elastic filterless optical networks. *IEEE/ACM Trans. Netw.*, 24(6):3578–3592, Dec. 2016.

[8] S. Ba, B. Chatterjee, S. Okamoto, N. Yamanaka, A. Fumagalli, and E. Oki. Route partitioning scheme for elastic optical networks with hitless defragmentation. *IEEE J. Opt. Commun. Netw.*, 8(6):356–370, Jun. 2016.

[9] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE J. Sel. Areas Commun.*, 14:903–908, Jun. 1996.

[10] A. Belbekkouche, M. Hasan, and A. Karmouch. Resource discovery and allocation in network virtualization. *IEEE Commun. Surveys Tuts.*, 14(4):1114–1128, Fourth 2012.

[11] T. Boffey. A note on minimal length hamilton path and circuit algorithms. *Opera. Res.*, 24:437–439, 1973.

[12] Bollobas. The chromatic number of random graphs. *Combinat.*, 8:49–55, 1988.

[13] J. Bondy and U. Murty. *Graph Theory*. Springer-Verlag London, 2008.

[14] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. WWW*, pages 107–117, Apr. 1998.

[15] P. Catlin. Supereulerian graphs: A survey. *Journal of Graph Theory*, 16:177–196, Jun. 1992.

[16] B. Chatterjee, N. Sarma, and E. Oki. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Commun. Surveys Tuts.*, 17:1776–1800, Third Quarter 2015.

[17] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang. Virtual network embedding through topology-aware node ranking. *SIGCOMM Comput. Commun. Rev.*, 41(2):38–47, Apr. 2011.

[18] S. Cho and S. Ramasubramanian. Localizing link failures in all-optical networks using monitoring tours. *Computer Networks*, 58(17):2–12, 2014.

[19] M. Chowdhury, M. Rahman, and R. Boutaba. ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Trans. Netw.*, 20(1):206–219, Feb. 2012.

[20] N. Chowdhury and R. Boutaba. A survey of network virtualization. *Computer Networks*, 54:862–876, Apr. 2010.

[21] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos. Elastic bandwidth allocation in flexible OFDM-based optical networks. *J. Lightw. Technol.*, 29:1354–1366, May. 2011.

[22] D. Du, K. Ko, and X. Hu. *Design and Analysis of Approximation Algorithms*. Springer-Verlag New York, 2012.

[23] Q. Duan, Y. Yan, and A. Vasilakos. A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *IEEE Transactions on Network and Service Management*, 9(4):373–392, Dec. 2012.

[24] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *J. Comput. Syst. Sci.*, 57:187 – 199, 1998.

[25] A. Fischer, J. Botero, M. Beck, H. Meer, and X. Hesselbach. Virtual network embedding: A survey. *IEEE Commun. Surveys Tuts.*, 15(4):1888–1906, Fourth 2013.

[26] M. Fok, Z. Wang, Y. Deng, and P. Prucnal. Optical layer security in fiber-optic networks. *IEEE Trans. Inf. Forensics Security*, 6:725–736, Sept. 2011.

[27] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combinat. Theory*, 16:47–56, Jan. 1974.

[28] L. Gong, H. Jiang, Y. Wang, and Z. Zhu. Novel location-constrained virtual network embedding LC-VNE algorithms towards integrated node and link mapping. *IEEE/ACM Trans. Netw.*, 24(6):3648–3661, Dec. 2016.

[29] L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu. Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks. *J. Opt. Commun. Netw.*, 5:836–847, Aug. 2013.

[30] L. Gong, X. Zhou, W. Lu, and Z. Zhu. A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks. *IEEE Commun. Lett.*, 16:1520–1523, Sept. 2012.

[31] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer, 1988.

[32] M. Halldorsson. A still better performance guarantee for approximate graph coloring. *Inform. Proc. Lett.*, 45:19–23, Jan. 1993.

[33] I. Houidi, W. Louati, W. Ameur, and D. Zeghlache. Virtual network provisioning across multiple substrate networks. *Computer Networks*, 55(4):1011–1023, Mar. 2011.

[34] Q. Hu, Y. Wang, and X. Cao. Resolve the virtual network embedding problem: A column generation approach. In *Proc. IEEE INFOCOM*, pages 410–414, Apr. 2013.

[35] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network. *IEEE Commun. Mag.*, 48:138–145, Aug. 2010.

[36] M. Jinno, H. Takara, and B. Kozicki. Dynamic optical mesh networks: Drivers, challenges and solutions for the future. In *Proc. of ECOC 2009*, pages 1–4, Sept. 2009.

[37] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *IEEE Commun. Mag.*, 47:66–73, Nov. 2009.

[38] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu. Research directions in network service chaining. In *Proc. IEEE SDN4FNS*, pages 1–7, Nov. 2013.

[39] D. Johnson and L. McGeoch. *The Traveling Salesman Problem: A Case Study in Local Optimization*. John Wiley and Sons, Ltd, 1997.

[40] M. Ju, F. Zhou, S. Xiao, and Z. Zhu. Power-efficient protection with directed p -cycles for asymmetric traffic in elastic optical networks. *J. Lightw. Technol.*, 34(17):4053–4065, Sept. 2016.

[41] J. Jue and G. Xiao. An adaptive routing algorithm for wavelengthrouted optical networks with a distributed control scheme. In *Proc. IEEE ICCCN*, pages 192–197, 2002.

[42] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer-Verlag Berlin Heidelberg, 2004.

[43] M. Klinkowski and K. Walkowiak. Routing and spectrum assignment in spectrum sliced elastic optical path network. *IEEE Commun. Lett.*, 15(8):884–886, Aug. 2011.

[44] B. Kozicki, H. Takara, T. Yoshimatsu, K. Yonenaga, and M. Jinno. Filtering characteristics of highly-spectrum efficient spectrum-sliced elastic optical path (SLICE) network. In *Proc. of OFC 2009*, pages 1–3, Mar. 2009.

[45] A. Leivadeas, C. Papagianni, and S. Papavassiliou. Efficient resource mapping framework over networked clouds via iterated local search-based request partitioning. *IEEE Trans. Parallel Distrib. Syst.*, 24(6):1077–1086, Jun. 2013.

[46] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu. Highly efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks. *IEEE Netw.*, 29:36–42, Sept./Oct. 2015.

[47] R. Mao, H. Xu, W. Wu, J. Li, Y. Li, and M. Lu. Overcoming the challenge of variety: big data abstraction, the next evolution of data management for AAL communication systems. *IEEE Commun. Mag.*, 53(1):42–47, Jan. 2015.

[48] M. Médard, D. Marquis, R. Barry, and S. Finn. Security issues in all-optical networks. *IEEE Netw.*, 11:42–48, Jun. 1997.

[49] B. Mukherjee. *Optical WDM Networks*. Berlin, Germany: SpringerVerlag, 2006.

[50] C. Papadimitriou and U. Vazirani. On two geometric problems related to the traveling salesman problem. *Journal of Algorithms*, 5:231–246, Jun. 1984.

[51] M. Rahman and R. Boutaba. SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization. *IEEE Transactions on Network and Service Management*, 10(2):105–118, Jun. 2013.

[52] R. Ramamurthy and B. Mukherjee. Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks. *IEEE/ACM Trans. Netw.*, 10(3):351–367, Jun. 2002.

[53] A. Razzaq, M. Hidell, and P. Din. Virtual network embedding: a hybrid vertex mapping solution for dynamic resource allocation. *Electrical and Computer Engineering*, Jan. 2012.

[54] D. Rosenkrantz, R. Stearns, and P. Lewis. Approximate algorithms for the traveling salesperson problem. In *IEEE Conference Record of 15th Annual Symposium on Switching and Automate Theory*, pages 33–42, Oct. 1974.

[55] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy. Network virtualization architecture: proposal and initial prototype. In *Proc. ACM SIGCOMM Visa*, pages 63–72, Aug. 2009.

[56] E. Scheinerman and D. Ullman. *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. Dover Publications, 2013.

[57] G. Shen, H. Guo, and S. Bose. Survivable elastic optical networks: survey and perspective. *Photonic Network Communications*, 31:71–87, Feb. 2016.

[58] L. Shi and S. Ólafsson. Nested partitions method for global optimization. *Opera. Res.*, 48:390–407, 2000.

[59] L. Shi, S. Ólafsson, and N. Sun. New parallel randomized algorithms for the traveling salesman problem. *Comput Opera. Res.*, 26:371–394, 1999.

[60] W. Shi, Z. Zhu, M. Zhang, and N. Ansari. On the effect of bandwidth fragmentation on blocking probability in elastic optical networks. *IEEE Trans. Commun.*, 61:2970–2978, Jul. 2013.

[61] S. Shirazipourazad, C. Zhou, Z. Derakhshandeh, and A. Sen. On routing and spectrum allocation in spectrum-sliced optical networks. In *Proc. INFOCOM 2013*, pages 385–389, Apr. 2013.

[62] N. Skorin-Kapov, J. Chen, and L. Wosinska. A new approach to optical networks security: Attack-aware routing and wavelength assignment. *IEEE/ACM Trans. Netw.*, 18:750–760, Jun. 2010.

[63] N. Skorin-Kapov, M. Furdek, S. Zsigmond, and L. Wosinska. Physical-layer security in evolving optical networks. *IEEE Commun. Mag.*, 54:110–117, Aug. 2016.

[64] S. Subramaniam and R. Barry. Wavelength assignment in fixed routing wdm networks. In *Proc. IEEE ICC*, pages 406–410, 1997.

[65] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, and G. Rouskas". Spectrum management techniques for elastic optical networks: A survey. *Optical Switching and Networking*, 13:34–48, Jul. 2014.

[66] I. Tomkos, E. Palkopoulou, and M. Angelou. A survey of recent developments on flexible/elastic optical networking. In *Proc. of ICTON*, pages 1–6, Jul. 2012.

[67] J. Turner and D. Taylor. Diversifying the internet. In *Proc. IEEE GLOBECOM*, volume 2, Dec. 2005.

[68] S. Turrini. Optimization in permutation spaces. *West. Res. Lab. Rep.*, 1996.

[69] K. Walkowiak, R. Goścień, M. Klinkowski, and M. Woźniak. Optimization of multicast traffic in elastic optical networks with distance-adaptive transmission. *IEEE Commun. Lett.*, 18(12):2117–2120, Dec. 2014.

[70] J. Wang, K. Wright, and K. Gopalan. Xenloop: a transparent high performance inter-vm network loopback. *Cluster Computing*, 12:141–152, 2008.

[71] Y. Wang, X. Cao, Q. Hu, and Y. Pan. Towards elastic and fine-granular bandwidth allocation in spectrum-sliced optical networks. *J. Opt. Commun. Netw.*, 4:906–917, Nov. 2012.

[72] Y. Wang, X. Cao, and Y. Pan. A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks. In *Proc. of INFOCOM*, pages 1503–1511, Apr. 2011.

[73] Z. Wang, Y. Han, T. Lin, H. Tang, and C. Song. Virtual network embedding by exploiting topological information. In *Proc. IEEE GLOBECOM*, pages 2603–2608, Dec. 2012.

[74] G. Wilfong and P. Winkler. Ring routing and wavelength translation. In *Proc. ACM-SIAM SODA 1988*, pages 333–341, 1988.

[75] H. Wu, F. Zhou, Z. Zhu, and Y. Chen. On the distance spectrum assignment in elastic optical networks. *IEEE/ACM Trans. Netw.*, 25(4):2391–2404, 2017.

[76] J. Yen. Finding the K shortest loopless paths in a network. *Manag. Sci.*, 17:712–716, Nov. 1971.

[77] M. Yu, Y. Yi, J. Rexford, and M. Chiang. Rethinking virtual network embedding: Substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38(2), Apr. 2008.

[78] H. Zang, J. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical networks magazine.*, pages 47–60, 2000.

[79] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proc. IEEE INFOCOM*, pages 594–602, Mar. 1996.

[80] M. Zhang, C. You, and Z. Zhu. On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks. *IEEE/ACM Trans. Netw., in Press*, pages 1–15, 2015.

[81] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein. Virtual network embedding with opportunistic resource sharing. *IEEE Trans. Parallel Distrib. Syst.*, 25(3):816–827, Mar. 2014.

[82] J. Zhu, B. Zhao, W. Lu, and Z. Zhu. Attack-aware service provisioning to enhance physical-layer security in multi-domain EONs. *J. Lightw. Technol.*, 34:2645–2655, Jun. 2016.

[83] Z. Zhu, W. Lu, L. Zhang, and N. Ansari. Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing. *J. Lightw. Technol.*, 31:15–22, Jan. 2013.

Acronyms

AcR	Acceptance Ratio
AI	Artificial Intelligence
BV-T	Bandwidth-Variable Transponders
BV-WSS	Bandwidth-Variable Wavelength-Selective Switches
DSA	Distance Spectrum Assignment
EON	Elastic Optical Network
FPGA	First Phase Greedy Algorithm
FS	Frequency Slot
GOF	Global Optimal Formulation
IoE	Internet of Everything
ILP	Integer Linear Programming
MDKP	Multi-Dimensional Knapsack Problem
MHP	Minimum Hamilton Path
MKP	Multiple Knapsack Problem
M2O	Many-to-One
MUFI	Maximum Used FS Index.
NN	Nearest Neighbor
NPM	Nested Partitions Method
ODSA	Ordered Distance Spectrum Assignment
O2O	One-to-One
POP	Permutation-based Optimization Problem
QoE	Quality of Experience
QoS	Quality of Service

QoT	Quality of Transmission
Rev	Revenue
RSA	Routing and Spectrum Assignment
RMSA	Routing, Modulation-level, and Spectrum Assignment
SA	Spectrum Assignment
SL	Substrate Link
SN	Substrate Node
VL	Virtual Link
VN	Virtual Node
VNE	Virtual Network Embedding
VNR	Virtual Network Request
WA	Wavelength Assignment
WDAG	Weighted Directed Auxiliary Graph
WDM	Wavelength Division Multiplexing

List of Publications

International Journals:

[J1]: **Haitao Wu**, Fen Zhou, Zuqing Zhu and Yaojun Chen "On the Distance Spectrum Assignment in Elastic Optical Networks" *IEEE/ACM Transactions on Networking.*, 25(4): 2391-2404, Aug. 2017.

[J2]: Min Ju, Fen Zhou, Shilin Xiao and **Haitao Wu** "Leveraging Spectrum Sharing and Defragmentation to *p*-Cycle Design in Elastic Optical Networks" *IEEE Commun. Letters*, vol. 21, no. 3, pp. 508-511, Mar. 2017.

International Conferences:

[C1]: **Haitao Wu**, Fen Zhou, Zuqing Zhu, Yaojun Chen: "Security-and-InterferenceAware Distance Spectrum Assignment in Elastic Optical Networks" *The IEEE 21st European Conference on Network and Optical Communications (NOC 2016)*, pp1-6, Portugal, Jun. 2016 (post-deadline paper).

[C2]: **Haitao Wu**, Fen Zhou, Zuqing Zhu, Yaojun Chen "An Experimental Comparison of Routing and Spectrum Assignment Algorithms in Elastic Optical Networks" *The Springer International conference on NETwork Games, COntrol and OPtimization (Netgcoop 2016)*, pp1-10, Avignon, Nov. 2016.

Submissions:

[S1]: **Haitao Wu**, Fen Zhou, Zuqing Zhu and Yaojun Chen "How the Network Topology, Traffic Distribution, and Routing Scheme Impact on the Spectrum Usage in Elastic Optical Networks" *IEEE Transactions on Networking.*, Submitted, 2018

[S2]: **Haitao Wu**, Fen Zhou and Yaojun Chen "On Virtual Network Embedding: Paths and Cycles" *IEEE/ACM Transactions on Networking.* Major Revision, Apr. 2018.

[S3]: **Haitao Wu**, Fen Zhou, Zuqing Zhu and Yaojun Chen "Analysis Framework of RSA Algorithms in Cycle Elastic Optical Networks" *IEEE/OSA Journal of Lightwave Technology.*, Major Revision, 2018