Campus des Cézeaux
1 Rue de la Chebarde
63178 Aubière
France

Campus Pampulha
6627 avenue Presidente Antônio Carlos
31270-901 Belo Horizonte - MG
Brésil

On lower bounds for the spectrum width for the routing and spectrum assignment problem

Project report 3rd year

Gustavo FERRÃO FONSECA
Pedro Henrique FERNANDES DA SILVA

Responsables Annegret WAGLER and Hervé KERIVIN
Date : 03/04/2020

Université Clermont Auvergne
Institut Supérieur d'Informatique, de Modélisation et de leurs Applications
Filière F4 - Calcul et Modélisation Scientifiques

Aubière, April 2020

Acknowledgements

We would like to express our thanks to the various people who contributed to the smooth running of this project.

First of all, we would like to thank our project tutors, Annegret WAGLER and Herve KERIVIN, for their support throughout our project and for sharing their expertise. We also thank Christophe DUHAMEL and Philippe MAHEY orientations and their availability as our French coordinators of the BRAFITEC program since 2018.

We also want to thank Martín Gomez Ravetti, professor at the Federal University of Minas Gerais and Brazilian coordinator of the BRAFITEC program, for placing his trust in our skills and for having offered us this opportunity to come to France. In addition, we would like to express our gratitude to *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES* for their financial support.

Table of figures

Résumé

Le problème de routage et d'assignation de spectre est important dans la modélisation des réseaux optiques modernes. Un réseau dans le RSA peut apparaître de différentes manières, comme des chemins, des arbres, des cycles, des 1-arbres et le cas général, chacun avec ses propres spécificités. Pour évaluer la qualité de toute solution proposée, il est possible d'utiliser une comparaison de borne inférieure. Nous allons établir une chaîne de bornes inférieures pour la largeur de spectre minimale et discuter de la taille des espaces entre les paramètres de cette chaîne. Nous allons également déterminer quels types de graphes minimaux non superfacts peuvent apparaître dans l'intersection pour chaque type de réseau.

Dans la séquence, nous présenterons un cadre proposé pour la génération de colonnes pour une formulation de arêtes-chemin pour le routage et l'attribution de spectre dans les réseaux optiques. être expliqué. Le cadre utilise une série de stratégies pour déterminer un routage et une attribution de spectre pour le problème en utilisant les solutions obtenues à partir des programmes linéaires.

Connaissant le framework, nous procédons à l'implémentation en langage C ++ en interface avec le solveur commercial CPLEX. Nous analysons ensuite les adaptations nécessaires pour pouvoir exécuter correctement le framework dans un environnement de calcul.

Le cadre a été testé avec des instances et les résultats montrent l'efficacité du cadre pour résoudre le problème RSA, ouvrant de nouvelles possibilités d'amélioration dans le cadre.

Mots-clés : **Problème de routage et d'attribution de spectre, graphes intersections des arêts, coloration d'intervalles, nombre chromatique d'intervalles, chaîne de bornes inférieures, superperfection.**

Abstract

The routing and spectrum assignment problem is important in modelling modern optical networks. A network in the RSA can appear in different ways, such as paths, trees, cycles, 1-trees and the general case, each one with its own specificities. To evaluate the quality of any proposed solution, it is possible to use a lower bound comparation. We are going to establish a chain of lower bounds for the minimum spectrum width, and discuss the size of the gaps between the parameters of this chain. We are also going to determine wich types of minimal non-superpefect graphs can appear in the intersection for each type of network.

In the sequence, we will present a proposed framework for column generation for an edge-path formulation for routing and spectrum assignment in optical networks. The framework uses two linear programs, the minimum cost multi commodity flow problem and the Edge-Path formulation, which will both be explained. The framework uses a series of strategies to determine a routing and spectrum assignment for the problem making use of the solutions obtained from the linear programs.

Knowing the framework we proceed to the implementation in C++ language in interface with the commercial solver CPLEX. Afterwards we analyze the necessary adaptations made to be able to correctly run the framework in an computational environment.

The framework was tested with instances and the results show the effectiveness of the framework to solve the RSA problem, opening new possibilities of improvement in the framework.

Keywords : **Routing and spectrum assignment problem, edge intersection graphs, interval coloring, interval chromatic number, chain of lower bounds, superperfection.**

Table of contents

Presentation of the project

This report describes a study which took place from October 2019 to the end of March 2020 at the *Institut Supérieur d'Informatique, de Modélisation et de leurs applications* (ISIMA) under the responsibility of Annegret Wagler and Hervé Kerivin . This work describes our 3$^{\text{rd}}$ year project at ISIMA.

This project was introduced to us in October, when the first meetings began. It has a part focused on the state of the art in which the problem of Routing and Spectrum Asignment is encountered which developed until December 2019. The defense of the theoretical part was made in early January.

The second part of the work is focused on how we can use the theoretical knowledge acquired to efficiently solve the Routing and Spectrum Assignment problem. The defense of this part of the work will be done in early April 2020.

Throughout the project we were accompanied by our tutors in different ways such as email exchanges, face-to-face meetings and online meetings. The frequency of contact varied during the project but throughout the period the project continued to be developed by the parties.

This report describes the work carried out. They are divided as follows : Initially the Routing and Spectrum Assignment problem is introduced. Then, we establish a chain of lower bounds for the solution and we present some concepts that will be useful to us.

Then the state of the art of the problem based on the scientific literature is made. In the next chapter we introduce the proposed method to try to solve the problem based on the lower bounds chain and the models that will be used during the framework.

Finally, we will present how we implemented the ideas and the results in the first tests that were done. It is important to emphasize that as a perspective we have continuity in the development of ideas and implementation that we started as well as tests in larger instances.

1 - Introduction

1.1 Routing and Spectrum Assignment problem

Flexgrid elastic optical networks constitute a new generation of optical networks in response to the sustained growth of data traffic volumes and demands in communication networks. In such networks, light is used as communication medium between sender and receiver nodes, and the frequency spectrum of an optical fiber is divided into narrow frequency slots of fixed spectrum width. Any sequence of consecutive slots can form a channel that can be switched in the network to create a lightpath (i.e., an optical connection represented by a route and a channel). The *routing and spectrum assignment (RSA) problem* consists of establishing the lightpaths for a set of end-to-end traffic demands, which involves finding a route and assigning a channel, i.e. an interval of consecutive frequency slots for each demand such that the intervals of lightpaths using a same edge in the network are disjoint, see e.g. [1]. Thereby, the following constraints need to be respected when dealing with the RSA problem :

1. *spectrum continuity* : the frequency slots allocated to a demand remain the same on all the edges of a route ;
2. *spectrum contiguity* : the frequency slots allocated to a demand must be contiguous ;
3. *non-overlapping spectrum* : a frequency slot can be allocated to at most one demand.

The RSA problem has started to receive a lot of attention over the last few years. It has been shown to be NP-hard [2, 3]. In fact, if for each demand the route is already known, the RSA problem reduces to the *spectrum assignment (SA) problem* and only consists of determining the demands' channels. The SA problem has been shown to be NP-hard on paths [4].

More precisely, we are given an optical network $G = (V, E)$ and a set $\mathcal{D}$ of end-to-end traffic demands between pairs o_k, d_k of nodes in G specifying the number w_k of required frequency slots to satisfy this demand. The routing consists of selecting, for each demand $k = (o_k, d_k, w_k) \in \mathcal{D}$, an (o_k, d_k)-path P_k as route from o_k to d_k through G ; we denote a routing of $\mathcal{D}$ by $\mathcal{P} = \{P_k : k \in \mathcal{D}\}$. The spectrum assignment consists of selecting, for each $k \in \mathcal{D}$, a channel S_k of w_k consecutive frequency slots such that the non-overlapping spectrum condition is satisfied along P_k ; we denote a spectrum assignment for $\mathcal{D}$ by $\mathcal{S} = \{S_k : k \in \mathcal{D}\}$.

Our objective is to study the minimum spectrum width $\chi_I(G, \mathcal{D})$ needed for any feasible solution of the RSA problem, given G and $\mathcal{D}$. For that, we will study a chain of lower bounds for $\chi_I(G, \mathcal{D})$, based on a reinterpretation of the RSA problem in combinatorial terms.

1.2 On lower bounds

In fact, the spectrum assignment can be interpreted as an interval coloring of the edge intersection graph $I(\mathcal{P})$ of the set $\mathcal{P}$ of selected routes :

— Each path $P_k \in \mathcal{P}$ becomes a node k of $I(\mathcal{P})$ and two nodes k and k' are joined by an edge if the corresponding paths $P_k, P_{k'}$ in G are in conflict as they share an edge (note : we do not care whether they share nodes).

— Any interval coloring in this graph $I(\mathcal{P})$ weighted with the demands w_k correctly solves the spectrum assignment : we assign a frequency interval S_k of w_k consecutive frequency slots (*spectrum contiguity*) to every node k of $I(\mathcal{P})$ (and, thus, to every path $P_k \in \mathcal{P}$ (*spectrum continuity*)) in such a way that the intervals of adjacent nodes are disjoint (*non-overlapping spectrum*).

Let $\mathbf{w} \in \mathbb{Z}_+^{|\mathcal{D}|}$ be the vector whose entries w_k are the slot requirements associated with the demands $k \in \mathcal{D}$. The interval chromatic number $\chi_I(I(\mathcal{P}), \mathbf{w})$ is the smallest size of a spectrum such that $I(\mathcal{P})$ weighted with the traffic demand w_k for each path P_k has a proper interval coloring. Given G and $\mathcal{D}$, the minimum spectrum width of any solution of the RSA problem, thus, equals

$$\chi_I(G, \mathcal{D}) = \min\{\chi_I(I(\mathcal{P}), \mathbf{w}) : \mathcal{P} \text{ possible routing of demands } \mathcal{D} \text{ in } G.\}$$

We are going to establish a chain of lower bounds for $\chi_I(G, \mathcal{D})$. Firstly, there are two lower bounds exclusively related to the routing aspect of the problem (not yet taking the spectrum assignment into account).

We denote by $s_R(G, \mathcal{D})$ the minimum number of slots that need to be installed on all edges of the optical network G to allow the routing of all demands in $\mathcal{D}$. The value $s_R(G, \mathcal{D})$ corresponds to the maximum edge load $w_{\mathcal{P}}(e) = \max(\sum_{P_k \ni e} w_k : e \in E)$ in the most balanced routing $\mathcal{P}$, i.e., to the minimum maximum edge load, taken over all possible routings :

$$s_R(G, \mathcal{D}) = \min(w_{\mathcal{P}}(e) : \mathcal{P} \text{ possible routing of demands } \mathcal{D} \text{ in } G).$$

Due to the non-overlapping spectrum condition, all channels S_k of paths routed along a same edge of G need to be disjoint, thus, $s_R(G, \mathcal{D})$ is a lower bound of $\chi_I(G, \mathcal{D})$. In Section 3.2, we discuss a way how to compute $s_R(G, \mathcal{D})$, as min-cost multi-commodity flow in an auxiliary network constructed from G.

We further consider the weighted clique number $\omega(I(\mathcal{P}), \mathbf{w})$ of the edge intersection graph $I(\mathcal{P})$ of the routing $\mathcal{P}$ (that is the maximum weight of a clique, a set of pairwise adjacent nodes, in $I(\mathcal{P})$, taking the node weights $\mathbf{w}$ into account). We denote by

$$\omega(G, \mathcal{D}) = \min\{\omega(I(\mathcal{P}), \mathbf{w}) : \mathcal{P} \text{ possible routing of demands } \mathcal{D} \text{ in } G\}$$

the *clique bound*, i.e., the minimum over all maximum weighted cliques in $I(\mathcal{P})$, taken over all possible routings $\mathcal{P}$.

The clique bound is sandwiched between $s_R(G, \mathcal{D})$ and $\chi_I(G, \mathcal{D})$. On the one hand, all paths in a routing $\mathcal{P}$ passing through a same edge e of G are mutually in conflict and form a clique in $I(\mathcal{P})$, which shows that the maximum edge load of the most balanced routing is a lower bound of $\omega(G, \mathcal{D})$. On the other hand, all channels S_k of paths P_k forcing a clique in $I(\mathcal{P})$ need to be disjoint due to the non-overlapping spectrum condition such that $\omega(I(\mathcal{P}), \mathbf{w}) \leq \chi_I(I(\mathcal{P}), \mathbf{w})$ holds for any $I(\mathcal{P})$ and, thus, $\omega(G, \mathcal{D})$ is a lower bound of $\chi_I(G, \mathcal{D})$.

Finally, we consider the weighted chromatic number of the edge intersection graph $I(\mathcal{P})$ of the routing $\mathcal{P}$. It is defined as the smallest number of colors needed to color every node k with w_k colors such that adjacent nodes receive different colors, and is denoted by $\chi(I(\mathcal{P}), \mathbf{w})$. We define

$$\chi(G, \mathcal{D}) = \min\{\chi(I(\mathcal{P}), \mathbf{w}) : \mathcal{P} \text{ possible routing of demands } \mathcal{D} \text{ in } G\}$$

as the *chromatic bound*, i.e., the minimum over all weighted chromatic numbers of $I(\mathcal{P})$, taken over all possible routings $\mathcal{P}$.

For any graph, the weighted chromatic number is sandwiched between the weighted clique number and the interval chromatic number. On the one hand, the nodes k of any clique Q need to receive at least $\sum_{k \in Q} w_k$ colors. On the other hand, we have the additional condition that in any interval coloring, the colors of any node need to be consecutive. Hence, we have $\omega(I(\mathcal{P}), \mathbf{w}) \leq \chi(I(\mathcal{P}), \mathbf{w}) \leq \chi_I(I(\mathcal{P}), \mathbf{w})$ for any $I(\mathcal{P})$. This implies that $\chi(G, \mathcal{D})$ is sandwiched between $\omega(G, \mathcal{D})$ and $\chi_I(G, \mathcal{D})$ such that we finally obtain the following chain of lower bounds for the minimum spectrum width

$$s_R(G, \mathcal{D}) \leq \omega(G, \mathcal{D}) \leq \chi(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D}). \tag{1.1}$$

There are instances of the RSA problem where there is a gap between any two parameters from this chain.

Example 1. Consider the following instance of the RSA problem with the optical network G shown in Fig. 1.1 and the following set $\mathcal{D}$ of demands :

k	$o_k \to d_k$	w_k	routing P_k
1	$a \to c$	1	$a \to b \to c$
2	$c \to e$	3	$c \to b \to d \to e$
3	$e \to f$	3	$e \to d \to f$
4	$f \to g$	3	$f \to d \to g$
5	$g \to h$	3	$g \to d \to h$
6	$h \to a$	3	$h \to d \to b \to a$

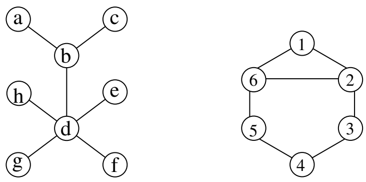

FIGURE 1.1: The network G and $I(\mathcal{P})$ of the routing used in Example 1.

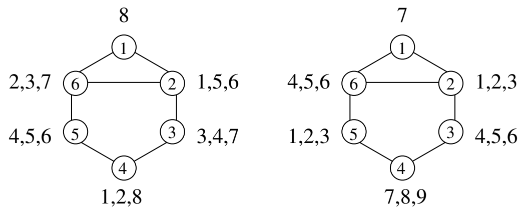

FIGURE 1.2: Minimal weighted coloring and interval coloring of $I(\mathcal{P})$ in Example 1.

As the network G is a tree, there is a unique routing $\mathcal{P}$ (as for every demand $k \in \mathcal{D}$, there is a unique path P_k for every origin/destination pair o_k, d_k), as indicated above.

Since the load of all edges incident to node d equals 6, $s_R(G, \mathcal{D}) = 6$ follows. The edge intersection graph $I(\mathcal{P})$ of the routing is also shown in Fig. 1.1. The nodes $1, 2, 6$ form a clique of weight 7, hence we have $\omega(G, \mathcal{D}) = 7$. An optimal weighted coloring of $I(\mathcal{P})$ using 8 colors is shown in Fig. 1.2, but any interval coloring of $I(\mathcal{P})$ needs at least 9 colors, see again Fig. 1.2. Hence, there are instances of the RSA problem with a gap between any two parameters from the chain (1.1).

1.3 Perfect and super-perfect graphs

We will see that $s_R(G, \mathcal{D})$ equals $\omega(G, \mathcal{D})$ only if all edge intersection graphs $I(\mathcal{P})$ have only cliques associated to subsets of paths in the routing $\mathcal{P}$ that all share a same edge of G, see Section 2.1.

A graph is *perfect* if and only if weighted clique number and weighted chromatic number coincide for all possible non-negative integral node weights [5]. Due to [6], perfect graphs are precisely the graphs without chordless cycles C_{2k+1} with $k \geq 2$, termed *odd holes*, or their complements, the *odd antiholes* $\overline{C}_{2k+1}$ (the complement $\overline{G}$ has the same nodes as G, but two nodes are adjacent in $\overline{G}$ if and only if they are non-adjacent in G). Hence, whether or not $\omega(G, \mathcal{D})$ equals $\chi(G, \mathcal{D})$ depends on the occurrence of non-perfect subgraphs (C_{2k+1} or $\overline{C}_{2k+1}$) in the edge intersection graphs $I(\mathcal{P})$.

Moreover, a graph is *superperfect* if and only if weighted clique number and interval chromatic number coincide for all possible non-negative integral node weights. In particular, every superperfect graph is perfect and perfect graphs G satisfy $\omega(G, \mathbf{w}) = \chi_I(G, \mathbf{w})$ for every $(0, 1)$-weighting $\mathbf{w}$ of its nodes (as in this case any interval coloring is a usual coloring). Whether or not $\omega(G, \mathcal{D})$ equals $\chi_I(G, \mathcal{D})$ depends on the occurrence of non-superperfect subgraphs in the edge intersection graphs $I(\mathcal{P})$.

In contrary to the case of perfect graphs, there is no complete characterization of minimal non-superperfect graphs known yet, but only several families.

A graph $G = (V, E)$ is *comparability* if and only if there exists a partial order $\mathcal{O}$ on $V \times V$ such that $uv \in E$ if and only if u and v are comparable w.r.t. $\mathcal{O}$. Comparability graphs form a subclass of superperfect graphs [7], but there are also superperfect non-comparability graphs such as e.g. even antiholes [8].

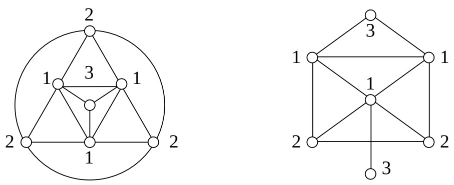

FIGURE 1.3: The graphs $\overline{A}_1$ and $\overline{A}_2$.

A complete list of minimal non-comparability graphs is presented in [9], the question which minimal non-comparability graphs are superperfect has been addressed in [10]. The minimal non-comparability graphs which are not superperfect by [10] are thus minimal non-superperfect [1] : the graphs $\overline{A}_1$, $\overline{A}_2$ shown in Fig. 1.3 and all

— odd holes C_{2k+1} and odd antiholes $\overline{C}_{2k+1}$ for $k \geq 2$,

— the graphs J_k for $k \geq 2$ and J'_k for $k \geq 3$ (see Fig. 1.4),

— the complements of D_k for $k \geq 2$ and of E_k, F_k for $k \geq 1$ (see Fig. 1.5).

Another infinite family of new minimal non-superperfect graphs containing even antiholes (and thus, a minimal non-comparability superperfect proper subgraph) has been presented in [11]. Let $\overline{C}_{2k}$ be an even antihole for some $k \geq 3$ and let $\overline{C}_{2k,j}$ be the graph obtained from $\overline{C}_{2k}$ by adding two adjacent nodes x and y, where x is adjacent to all nodes of $\overline{C}_{2k}$ but 1 and 2, and y is adjacent to all nodes of $\overline{C}_{2k}$ but $2 + j$ and $3 + j$, for all even j with $0 \leq j < k$. Note that $\overline{F}_1$ can

1. It has been noticed in [11] that Andreae [10] wrongly determined $\overline{A}_2$ as superperfect which is, in fact, not the case (see Fig. 1.3 for a weight vector $\mathbf{w}$ showing that $\omega(\overline{A}_2, \mathbf{w}) = 5 < 6 = \chi_I(\overline{A}_2, \mathbf{w})$ holds), and wrongly determined J'_2 as non-superperfect (see the proof of its superperfection presented in [11]).

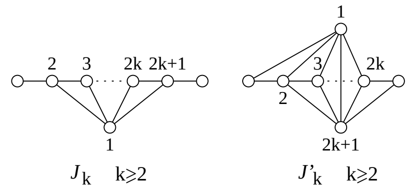

FIGURE 1.4: Minimal non-superperfect graphs : J_k for $k \geq 2$, J'_k for $k \geq 3$.

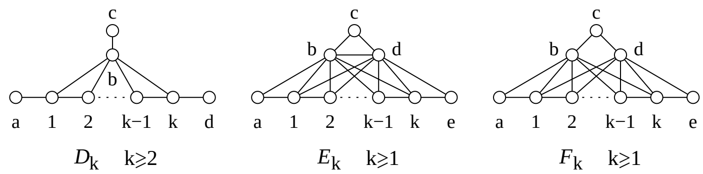

FIGURE 1.5: Minimal non-superperfect graphs : the complements of D_k, E_k, F_k.

be considered to be $\overline{C}_{4,0}$; the graphs $\overline{C}_{6,0}$ and $\overline{C}_{6,2}$ are shown in Fig. 1.6 together with weight vectors causing a gap between weighted clique and interval chromatic number.

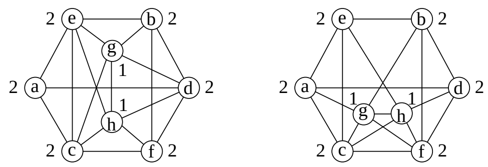

FIGURE 1.6: Minimal non-superperfect graphs $\overline{C}_{6,0}$ and $\overline{C}_{6,2}$ containing $\overline{C}_6$.

Note that we have $\omega(G, \mathbf{1}) < \chi_I(G, \mathbf{1})$ with $\mathbf{1} = (1, \ldots, 1)$ if G is an odd hole or an odd antihole (as they are not perfect by [6]), whereas we have for perfect non-superperfect graphs that $\omega(G, \mathbf{w}) < \chi_I(G, \mathbf{w})$ is attained for some $\mathbf{w} \neq \mathbf{1}$.

We examine, for different underlying networks G, the questions

— which subsets of paths in $\mathcal{P}$ give raise to cliques in $I(\mathcal{P})$,

— how hard it is to compute $\omega(G, \mathcal{D})$, and

— when there is a solution of the RSA problem with $\omega(G, \mathcal{D})$ as spectrum width (which depends on the occurrence of non-superperfect graphs in $I(\mathcal{P})$).

For some networks G, the edge intersection graphs form well-studied graph classes : if G is a path (resp. tree, resp. cycle), then $I(\mathcal{P})$ is an interval graph (resp. EPT graph, resp. circular-arc graph). In all these cases, it is known that $\omega(I(\mathcal{P}), \mathbf{w}))$ can be computed in polynomial time [8, 12, 13]. However, if G is a sufficiently large grid, then it is known by [14] that $I(\mathcal{P})$ can be any graph (and computing $\omega(I(\mathcal{P}), \mathbf{w})$) is NP-hard).

Modern optical networks do not fall in any of these classes, but are 2-connected, sparse planar graphs with small maximum degree. We study this case and exhibit many different ways to build cliques in $I(\mathcal{P})$ (which suggests that determining $\omega(I(\mathcal{P}), \mathbf{w})$) is NP-hard). In [11] it was characterized which of the minimal non-comparability non-superperfect graphs occur in $I(\mathcal{P})$, depending on the underlyong networks. It turned out that all such non-superperfect graphs can occur as soon as the optical networks satisfy minimal survivability conditions concerning edge or node failures (which implies that there is not always a solution of the RSA problem with $\omega(G, \mathcal{D})$ as spectrum width).

In order to estimate the gap between $\omega(G, \mathcal{D})$ and $\chi_I(G, \mathcal{D})$, we extend the concept of χ-binding functions introduced in [15] for usual coloring to interval coloring in weighted graphs, that is, to χ_I-binding functions f with

$$\chi_I(I(\mathcal{P}), w) \leq f(\omega(I(\mathcal{P}), w))$$

for edge intersection graphs $I(\mathcal{P})$ in a certain class of networks and all possible non-negative integral weights w.

2 - State of the art

2.1 If the network is a path

If the optical network is a path P, then there exists exactly one (o_k, d_k)-path P_k in P for every traffic demand between a pair o_k, d_k of nodes. Hence, if P is a path, then $\mathcal{P}$ and $I(\mathcal{P})$ are uniquely determined for any set $\mathcal{D}$ of demands, and the RSA problem reduces to the spectrum assignment part. The edge intersection graph $I(\mathcal{P})$ of the (unique) routing $\mathcal{P}$ of the demands is an *interval graph* (i.e. the intersection graph of intervals in a line, here represented as subpaths of a path).

Cliques in edge intersection graphs of paths in a path. It is well-known that cliques in interval graphs $I(\mathcal{P})$ correspond to subsets of paths in $\mathcal{P}$ having one edge in common. For any edge e of the network G, let

$$\mathcal{P}(e) = \{P \in \mathcal{P} : e \in P\}$$

denote the subset of paths in $\mathcal{P}$ containing edge e. Then $\mathcal{P}(e)$ corresponds to a clique in $I(\mathcal{P})$, called an *edge-clique*.

In interval graphs $I(\mathcal{P})$, we clearly have only edge-cliques and, thus, the maximum edge load of the routing $\mathcal{P}$ in the network corresponds to the maximum weighted clique in the edge intersection graph $I(\mathcal{P})$ which implies

$$s_R(G, \mathcal{D}) = \omega(I(\mathcal{P}), \mathbf{w}) = \omega(G, \mathcal{D}).$$

Note that the weighted clique number of an interval graph can be computed in polynomial time by [5] and, thus, also $\omega(G, \mathcal{D})$.

Perfection of edge intersection graphs of paths in a path. Since the edge intersection graphs of paths in a path are interval graphs, they are perfect by [16]. Thus, we have $\omega(I(\mathcal{P}), \mathbf{w}) = \chi(I(\mathcal{P}), \mathbf{w})$ for the unique edge intersection graph and all node weights $\mathbf{w}$. This implies :

Corollary 1. *If the network G is a* path*, then $\omega(G, \mathcal{D}) = \chi(G, \mathcal{D})$ holds for any set $\mathcal{D}$ of demands.*

Both parameters can be computed in polynomial time by [5], see above.

Superperfection of edge intersection graphs of paths in a path. It was examined in [11] which of the minimal non-comparability non-superperfect graphs can occur in such edge intersection graphs :

Theorem 2.1.1 (Kerivin and Wagler [11]). *If $\mathcal{P}$ is a set of paths in a* path, *then $I(\mathcal{P})$ can contain the graphs J_k for all $k \geq 2$, J'_k for all $k \geq 3$ and $\overline{E}_2$, but none of the other minimal non-comparability non-superperfect graphs.*

In addition, we have the following conditions to *interval graphs* :

Theorem 2.1.2 (Lekkerkerker and Boland [17]). *A graph is a interval graph if and only if it contains no induced bipartite-claw, umbrella, n-net for any $n \geq 2$, n-tent for any $n \geq 3$ or C_n for any $n \geq 4$.*

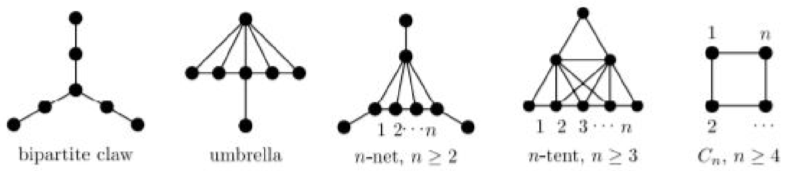

FIGURE 2.1: Minimal forbidden induced subgraphs for the class of interval graphs. [18]

This implies that even edge intersection graphs of paths in a path are not necessarily superperfect. For any (minimal) non-superperfect interval graph $I(\mathcal{P})$, there is a weight vector $\mathbf{w} \neq \mathbf{1}$ such that $\omega(I(\mathcal{P}), \mathbf{w}) < \chi_I(I(\mathcal{P}), \mathbf{w})$. Stockmeyer showed (see [8]) that determining the interval chromatic number of an interval graph is NP-hard, even if the weights $\mathbf{w}$ are restricted to be $(1, 2)$-valued only.

We next study how large the gap between weighted clique number and interval chromatic number can be in the worst case.

Lemma 1 (Argiroffo and Wagler). *Consider $\overline{E}_2$ with weight vector $\mathbf{w} = (3g, g, 2g, g, 2g, g, 3g)$ for some $g \in \mathbb{N}$. We have $\omega(\overline{E}_2, \mathbf{w}) = 4g$ and $\chi_I(\overline{E}_2, \mathbf{w}) = 5g$.*

Lemma 2 (Argiroffo and Wagler). *Consider the graphs J_k for all even $k \geq 2$ with weights*

$$w_i = \begin{cases} g & \textit{if } 1 \leq i \leq k+1 \textit{ and } i \textit{ is odd or } k+2 \leq i \leq 2k \textit{ and } i \textit{ is even} \\ 2g & \textit{otherwise} \end{cases}$$

for some $g \in \mathbb{N}$. We have $\omega(J_k, \mathbf{w}) = 4g$ and $\chi_I(J_k, \mathbf{w}) = 5g$.

Hence, the ratio of $\omega(I(\mathcal{P}), \mathbf{w})$ and $\chi_I(I(\mathcal{P}), \mathbf{w})$ taken over all weights and all interval graphs is at least $5/4$.

Summary for edge intersection graphs of paths in a path. We can summarize the relations of the lower bounds for the interval chromatic number as

$$\omega(I(\mathcal{P}), \mathbf{w}) = \chi(I(\mathcal{P}), \mathbf{w}) \leq \chi_I(I(\mathcal{P}), \mathbf{w}) \leq \frac{5}{4}\omega(I(\mathcal{P}), \mathbf{w})$$

and, thus, for the minimum spectrum width by

$$s_R(G, \mathcal{D}) = \omega(G, \mathcal{D}) = \chi(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D}) \leq \frac{5}{4}\omega(G, \mathcal{D}).$$

Problem 1.
- Are there other minimal non-superperfect interval graphs different from J_k for all $k \geq 2$, J'_k for all $k \geq 3$ and $\overline{E}_2$?
- Are there other (not necessarily minimal) non-superperfect interval graphs G and weights $\mathbf{w}$ with
$$\frac{\omega(G, \mathbf{w})}{\chi_I(G, \mathbf{w})} > \frac{5}{4}?$$

2.2 If the network is a tree

If the optical network is a tree T, then there is also exactly one (u, v)-path P_{uv} in T for every traffic demand between a pair u, v of nodes. Hence, if T is a tree, then $\mathcal{P}$ and $I(\mathcal{P})$ are uniquely determined for any set of demands, and the RSA problem again reduces to the spectrum assignment part. The edge intersection graph $I(\mathcal{P})$ of the (unique) routing $\mathcal{P}$ is called an *EPT graph.* We recall results from [13] on cliques and holes in EPT graphs and from Kerivin and Wagler [11] on the minimal non-comparability, non-superperfect graphs that can occur in EPT graphs. We further examine the possible gaps between the studied parameters $s_R(G, \mathcal{D})$, $\omega(G, \mathcal{D})$, $\chi(G, \mathcal{D})$ and $\chi_I(G, \mathcal{D})$ as well as bounds for the resulting χ_I-binding functions.

Cliques in edge intersection graphs of paths in a tree. Golumbic and Jamison showed in [13] that there exist two types of path configurations in a tree that give raise to cliques in the corresponding edge intersection graphs.

For any claw $K = K_{1,3}$ in T, let

$$\mathcal{P}(K) = \{P \in \mathcal{P} : P \text{ contains two edges of } K\},$$

then $\mathcal{P}(K)$ clearly corresponds to a clique in $I(\mathcal{P})$, called a *claw-clique.* For illustration, see a path representation of $\overline{D}_2$ in Fig. 2.2.

Golumbic and Jamison showed that, besides edge-cliques, there are no further path configurations in a tree implying cliques in the edge intersection graphs :

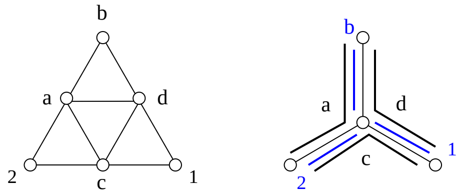

FIGURE 2.2: The graph $\overline{D}_2 = I(\mathcal{P})$ with $\mathcal{P}$ in a tree ; $1cd$, $2ac$ and abd are edge-cliques, acd is a claw-clique.

Theorem 2.2.1 (Golumbic and Jamison [13]). *Any maximal clique of the edge intersection graph of paths in a tree T corresponds to an edge-clique $\mathcal{P}(e)$ for some edge e of T or to a claw-clique $\mathcal{P}(K)$ for some claw $K = K_{1,3}$ in T.*

As a consequences, the problem of finding a clique of maximum cardinality in the edge intersection graph of paths in a tree can be solved in polynomial time by inspecting all edges and all claws in that tree [13]. This clearly carries over to the weighted version of the problem :

Corollary 2. *If $I(\mathcal{P})$ is the edge intersection graph of a collection $\mathcal{P}$ of paths in a tree, then the weighted clique number $\omega(I(\mathcal{P}), \mathbf{w})$ can be computed in polynomial time for any integral node weighting $\mathbf{w}$.*

Furthermore, recall that the routing is unique in trees such that $\mathcal{P}$ and $I(\mathcal{P})$ are uniquely determined for any set of demands. The weighted clique number $\omega(I(\mathcal{P}), \mathbf{w})$ of this unique edge intersection graph provides us a lower bound for the minimum spectrum width of any interval coloring, and can be computed in polynomial time. However, the possible presence of claw-cliques in $I(\mathcal{P})$ shows that we might have

$$s_R(G, \mathcal{D}) < \omega(I(\mathcal{P}), \mathbf{w}) = \omega(G, \mathcal{D}).$$

For illustration, take the path representation of $\overline{D}_2$ in Fig. 2.2 together with the following node weights :

$$w_v = \begin{cases} 1 & \text{if } v \in \{1, 2, b\} \\ 2 & \text{if } v \in \{a, c, d\} \end{cases}$$

Then we get a maximum edge load of 5 but $\omega(\overline{D}_2, \mathbf{w}) = 6$ due to the claw-clique composed of a, c, d.

In order to estimate how large this gap can be in general, we show :

Lemma 3 ([11]). *For the edge intersection graph $I(\mathcal{P})$ of paths $\mathcal{P}$ in a tree, we have $\omega(G, \mathcal{D}) \leq \frac{3}{2} s_R(G, \mathcal{D})$.*

Démonstration. Considering Theorem 2.2.1 [13], the ratio $\omega(G, \mathcal{D}) \leq \frac{p}{q} s_R(G, \mathcal{D})$ can be determined by the analysis of the ratio in the claw-clique case, as the ratio in the edge-clique case is trivial. Let P_a, P_b, P_c the paths composing the claw-clique, each one with weight $\mathbf{w}$. In this case we can consider

$$s_R \;=\; max\,\{w_a + w_b\ ,\ w_a + w_c\ ,\ w_b + w_c\} \tag{2.1}$$

In the other hand, we can consider also

$$\omega \;=\; w_a + w_b + w_c \tag{2.2}$$

With equations 2.1 and 2.2, we can conclude that the ratio between ω and s_R is maximal if $w_a = w_b = w_c$. Finally, let $w_a = w_b = w_c = q$ and we have $\omega / s_R = 3/2$

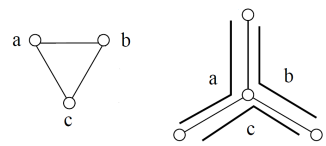

FIGURE 2.3: $I(\mathcal{P})$ graph with $\mathcal{P}$ in a clic ; *abc* is a claw-clique.

Perfection of edge intersection graphs of paths in a tree. It is known from [13] that edge intersection graphs $I(\mathcal{P})$ of paths in a tree are not necessarily perfect as they can contain odd holes. More precisely, Golumbic and Jamison showed the following :

Theorem 2.2.2 (Golumbic and Jamison [13]). *If the edge intersection graph $I(\mathcal{P})$ of a collection $\mathcal{P}$ of paths in a tree T contains a hole C_k with $k \geq 4$, then T contains a star $K_{1,k}$ with nodes $b, a_1, \ldots, a_k$ and there are k paths $P_1, \ldots, P_k$ in $\mathcal{P}$ such that P_i precisely contains the edges ba_i and ba_{i+1} of this star (where indices are taken modulo k).*

Figure 2.4 illustrates the case of $C_5 = I(\mathcal{P})$.

From the result above, Golumbic and Jamison deduced the possible adjacencies of a hole which further implies that several graphs cannot occur as induced subgraphs of EPT graphs, including the complement of the P_6. This implies particularly that no antihole $\overline{C}_k$ for $k \geq 7$ can occur in such graphs :

Theorem 2.2.3 (Golumbic and Jamison [13]). *The edge intersection graph of paths in a tree is perfect if and only if it does not contain an odd hole.*

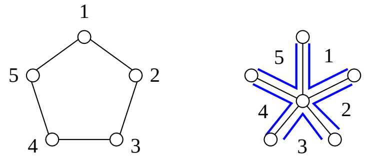

FIGURE 2.4: The odd hole $C_5 = I(\mathcal{P})$ with $\mathcal{P}$ in a star.

Note that the edge intersection graph of paths in a tree with maximum degree 4 is in any case perfect by Theorem 2.2.1. This implies :

Corollary 3. *For the edge intersection graph $I(\mathcal{P})$ of paths $\mathcal{P}$ in a tree T, we have $\omega(G, \mathcal{D}) \leq \chi(G, \mathcal{D})$ and equality holds if T has maximum degree at most 4.*

Superperfection of edge intersection graphs of paths in a tree. It was examined in [11] which of the minimal non-comparability non-superperfect graphs can occur in such edge intersection graphs :

Theorem 2.2.4 ([11]). *If $\mathcal{P}$ is a set of paths in a* tree*, then $I(\mathcal{P})$ can contain $\overline{A}_1$, $\overline{A}_2$ and*

- *odd holes C_{2k+1} for $k \geq 2$, but no odd antiholes $\overline{C}_{2k+1}$ for $k \geq 3$,*
- *the graphs J_k for all $k \geq 2$ and J'_k for all $k \geq 3$,*
- *the graphs $\overline{D}_2$, $\overline{D}_3$, $\overline{E}_1$, $\overline{E}_2$, $\overline{E}_3$, $\overline{F}_1$, $\overline{F}_2$, $\overline{F}_3$, but none of $\overline{D}_k$, $\overline{E}_k$, $\overline{F}_k$ for $k \geq 4$.*

Note that the proof for odd holes and odd antiholes follows from [13]. In [11], according path collections are presented for all other affirmative cases, and a $\overline{P}_6$ (which cannot occur in EPT graphs by [13]) is exhibited as common subgraph of the remaining cases.

This implies that perfect edge intersection graphs of paths in a tree are not necessarily superperfect.

Thus, whenever the underlying network is a tree, the weighted clique number $\omega(I(\mathcal{P}), \mathbf{w})$ of the edge intersection graph associated with the unique routing $\mathcal{P}$ provides us a lower bound for the minimum spectrum width of any interval coloring of $I(\mathcal{P})$, but we do not necessarily have equality as non-superperfect subgraphs in $I(\mathcal{P})$ can determine the spectrum width of an interval coloring. In order to examine the possible gap between weighted clique and interval chromatic number, we show :

Lemma 4. *If $I(\mathcal{P})$ is an odd hole, then $\chi_I(I(\mathcal{P}), \mathbf{w}) \leq \frac{3}{2}\omega(I(\mathcal{P}), \mathbf{w})$ for all non-negative integral weights $\mathbf{w}$.*

Note that the worst case is attained if all weights in $\mathbf{w}$ are equal.

Summary for edge intersection graphs of paths in a tree. We can summarize the relations of the lower bounds for the interval chromatic number as

$$\omega(I(\mathcal{P}),\mathbf{w}) \leq \chi(I(\mathcal{P}),\mathbf{w}) \leq \chi_I(I(\mathcal{P}),\mathbf{w}) \leq f(\omega(I(\mathcal{P}),\mathbf{w}))$$

and, thus, for the minimum spectrum width by

$$s_R(G,\mathcal{D}) \leq \omega(G,\mathcal{D}) \leq \chi(G,\mathcal{D}) \leq \chi_I(G,\mathcal{D}) \leq \frac{3}{2}\omega(G,\mathcal{D}) \leq f(\omega(G,\mathcal{D})).$$

Example 1 shows that there can be a gap between any two parameters from this chain.

Problem 2.
- Are there other minimal non-superperfect EPT graphs different from the ones cited in Theorem 2.2.4 ?
- If there are examples of trees with $s_R(G,\mathcal{D}) \leq \frac{2}{3}\omega(G,\mathcal{D})$ and other examples with $\omega(G,\mathcal{D}) \leq \frac{2}{3}\chi_I(G,\mathcal{D})$, are there cases with $s_R(G,\mathcal{D}) \leq \frac{4}{9}\chi_I(G,\mathcal{D})$ or worse (some combination of the two cases below) ?

2.3 If the network is a cycle

If the network is a cycle C, then there exist exactly two (u,v)-paths P_{uv} in C for every traffic demand between a pair u,v of nodes. Hence, if C is a cycle, then the number of possible routings $\mathcal{P}$ (and their edge intersection graphs $I(\mathcal{P})$) is exponential in the number $|\mathcal{D}|$ of end-to-end traffic demands, namely $2^{|\mathcal{D}|}$.

Moreover, the edge intersection graphs of paths in a cycle are clearly *circular-arc graphs* (that are the intersection graphs of arcs in a cycle, here represented as paths in a hole C_n).

We recall results on cliques in circular-arc graphs and which minimal non-comparability, non-superperfect graphs can occur in such graphs. We further examine the possible gaps between the studied parameters $s_R(G,\mathcal{D})$, $\omega(G,\mathcal{D})$, $\chi(G,\mathcal{D})$ and $\chi_I(G,\mathcal{D})$ as well as bounds for the resulting χ_I-binding functions.

Cliques in edge intersection graphs of paths in a cycle. It is well-known that there are only two types of cliques in a circular-arc graph $I(\mathcal{P})$: subsets $\mathcal{P}'$ of paths in a cycle $C = (V,E)$ that all share a same edge $e \in E$ (again called edge-cliques), or pairwise intersect in some edge and whose union covers all edges in E, but whose intersection is empty, i.e.,

$$E(P) \cap E(P') \neq \emptyset \ \forall P, P' \in E, \ \cup_{P\in\mathcal{P}'} E(P) = E, \ \cap_{P\in\mathcal{P}'} E(P) = \emptyset,$$

that we call *cycle-cliques*. For illustration, see a path representation of $\overline{D}_2$ in Fig. 2.5.

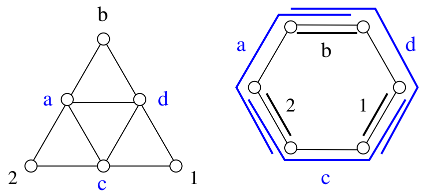

FIGURE 2.5: The graph $\overline{D}_2 = I(\mathcal{P})$ with $\mathcal{P}$ in a cycle ; $1cd$, $2ac$ and abd are edge-cliques, acd is a cycle-clique.

Remark 1. $I(\mathcal{P})$ is a *Helly circular-arc graph* if and only if the paths in $\mathcal{P}$ corresponding to a clique of $I(\mathcal{P})$ have non-empty intersection, i.e., Helly circular-arc graph (introduced by Gavril [19]) have edge-cliques only. $\overline{D}_2$ is not a Helly circular-arc graph.

A maximum cardinality clique in a circular-arc graph G can be found in polynomial time by a procedure proposed by Gavril [12]. The argument is based on the following facts. For each node $v \in V(G)$, the closed neighborhood $N(v) \cup \{v\}$ contains a subgraph G_v (induced by v and all its neighbors that correspond to paths containing the endnodes of the path of v). This subgraph G_v is the complement of a bipartite graph (whose clique number $\omega(G_v)$ can be determined in polynomial time). Every maximum cardinality clique in G is contained in one of the subgraphs G_v such that

$$\omega(G) = \max\{\omega(G_v) : v \in V(G)\}$$

for any circular-arc graph G can be determined in polynomial time. This clearly carries over to the weighted version of the problem and we obtain :

Corollary 4. *If $I(\mathcal{P})$ is the edge intersection graph of paths $\mathcal{P}$ in a cycle, then the weighted clique number $\omega(I(\mathcal{P}), \mathbf{w})$ can be computed in polynomial time for any non-negative integral node weighting $\mathbf{w}$.*

However, as the number of possible routings $\mathcal{P}$ is exponential in the number of given traffic demands in a cycle network C, determining the lower bound

$$\omega(G, \mathcal{D}) = \min\{\omega(I(\mathcal{P}), w) : \mathcal{P} \text{ possible routing in } C\}$$

for the spectrum width of any interval coloring is clearly NP-hard to compute. Moreover, the possible presence of cycle-cliques in $I(\mathcal{P})$ shows that we might have $s_R(G, \mathcal{D}) < \omega(G, \mathcal{D})$. For illustration, take the path representation of $\overline{D}_2$ in Fig. 2.5 together with the following node weights :

$$w_v = \begin{cases} 1 & \text{if } v \in \{1, 2, b\} \\ 2 & \text{if } v \in \{a, c, d\} \end{cases}$$

Then we get a maximum edge load of 5 but $\omega(\overline{D}_2, \mathbf{w}) = 6$ due to the cycle-clique composed of a, c, d.

Perfection of edge intersection graphs of paths in a cycle. It is well-known that circular-arc graphs are not necessarily perfect as they can contain both odd holes and odd antiholes, see e.g. [18] and Fig. 2.6 for illustration.

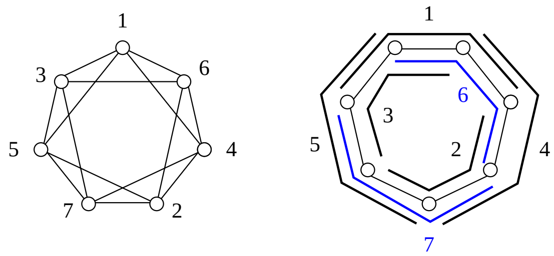

FIGURE 2.6: The graph $\overline{C}_7 = I(\mathcal{P})$ with $\mathcal{P}$ in a cycle.

Superperfection of edge intersection graphs of paths in a cycle. In order to address the question which of the studied perfect minimal non-comparability, non-superperfect graphs can occur in circular-arc graphs, we either present according path collections for the affirmative cases or exhibit a minimal non-circular-arc graph otherwise.

From the Theorem 2.1.2, is shown in [18] how to deduce some minimal forbidden induced subgraphs for the class of circular-arc graphs as follows. "Given a minimal forbidden induced subgraph H for the class of interval graphs, if H is a non-circular-arc graph, then H is minimally non-circular-arc. Otherwise, if H is a circular-arc graph, then $H \cup K_1$ is a minimally non-circular-arc graph, and furthermore all disconnected minimally non-circular-arc graphs are obtained this way. Since the umbrella, net, n-tent for all $n \geq 3$, and C_n for all $n \geq 4$ are circular-arc graphs, but the bipartite claw and n-net for all $n \geq 3$ are not, this observation and Theorem 2.1.2 lead to the following result."

Corollary 5 ([20]). *The following graphs are minimally non-circular-arc graphs : bipartite claw, net* $\cup K_1$*, n-net for all* $n \geq 3$*, umbrella* $\cup K_1$*,* $n - tent \cup K_1$ *for all* $n \geq 3$*, and* $C_n \cup K_1$ *for every* $n \geq 4$*. Any other minimally non-circular-arc graph is connected.*

The graphs presented in the Corollary 5 are called basic minimally non-circular-arc graphs by [21] and any other is called non-basic. So we can conclude form the Theorem 2.1.2 and the Corollary 5 it is possible to deduce the structural property for all non-basic minimally non-circular-arc graph as follows.

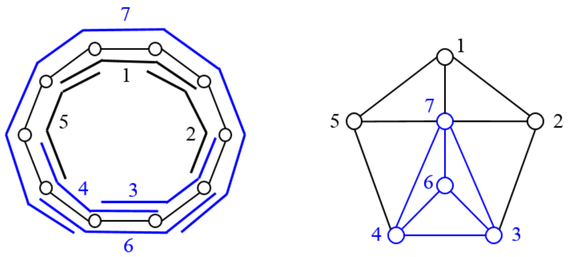

FIGURE 2.7: $\mathcal{P}$ in a cycle and $I(\mathcal{P})$

.

Corollary 6 ([21]). *). If G is a non-basic minimally non-circular-arc graph, then G has an induced subgraph H that is isomorphic to an umbrella, a net, a j-tent for some $j \geq 3$, or C_j for some $j \geq 4$. In addition, each vertex v of $G - H$ is adjacent to at least one vertex of H.*

Then, we can present the possibles minimally circular arc graphs as follows.

Theorem 2.3.1. *If $\mathcal{P}$ is a set of paths in a* cycle*, then* circular-arc graph $I(\mathcal{P})$ *can contain $\overline{A}_1$ but not $\overline{A}_2$ and*

- *all odd holes C_{2k+1} and odd antiholes $\overline{C}_{2k+1}$ for $k \geq 2$,*
- *the graphs J_k for all $k \geq 2$ and J'_k for all $k \geq 3$,*
- *$\overline{D}_2$, $\overline{D}_3$, $\overline{D}_4$, but not the graphs $\overline{D}_k$ for $k \geq 5$,*
- *$\overline{E}_1$ and $\overline{E}_2$, but not the graphs $\overline{E}_k$ for $k \geq 3$,*
- *$\overline{F}_2$, but not $\overline{F}_1$ neither the graphs $\overline{F}_k$ for $k \geq 3$.*

Summary for edge intersection graphs of paths in a cycle. We can summarize the relations of the lower bounds for the interval chromatic number as

$$\omega(I(\mathcal{P}), \mathbf{w}) \leq \chi(I(\mathcal{P}), \mathbf{w}) \leq \chi_I(I(\mathcal{P}), \mathbf{w}) \leq f(\omega(I(\mathcal{P}), \mathbf{w}))$$

and, thus, for the minimum spectrum width by

$$s_R(G, \mathcal{D}) \leq \omega(G, \mathcal{D}) \leq \chi(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D}) \leq \frac{3}{2}\omega(G, \mathcal{D}) \leq f(\omega(G, \mathcal{D})).$$

The following example 1 shows that there can be a gap between any two parameters from this chain.

Example 1. **(Wagler)** Consider the following instance with weight vector $\mathbf{w} = (3g, 3g, 3g, 3, 3g, g, g)$ for some $g \in \mathbb{N}$. We have $s_R = 7g$ and $\omega = 8g$ as the nodes $3, 4, 6, 7$ form a clique of weight $8g$.

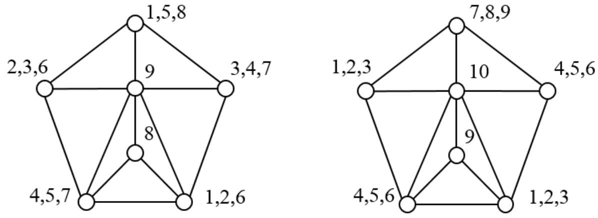

FIGURE 2.8: Minimal weighted coloring and interval coloring of $I(\mathcal{P})$ in Example 1.

An optimal weighted coloring of $I(\mathcal{P})$ using $9g$ colors is shown in Fig. 2.8, but any interval coloring of $I(\mathcal{P})$ needs at least 10 colors, see again Fig. 2.8. Hence, there are instances of the RSA problem in a cycle with a gap between any two parameters from the chain.

Problem 3.

— Are there other minimal non-superperfect circular-arc graphs different from the ones cited in Theorem 2.3.1 ?

— If there are examples of cycles with $s_R(G, \mathcal{D}) \leq \frac{2}{3}\omega(G, \mathcal{D})$ and other examples with $\omega(G, \mathcal{D}) \leq \frac{2}{3}\chi_I(G, \mathcal{D})$, are there cases with $s_R(G, \mathcal{D}) \leq \frac{4}{9}\chi_I(G, \mathcal{D})$ (some combination of the two cases below) ?

2.4 If the network is a 1-tree

Modern optical networks have clearly not a tree-like structure due to survivability aspects concerning node or edge failures in the network G. At least the subset of "core nodes" has to lie on a cycle (to have the choice between two paths, see e.g. [22]). We wonder which cliques and which minimal non-superperfect graphs from the list in [9] can occur in edge-intersection graphs of paths in *1-trees* (that are graphs obtained from a tree by adding one edge, i.e., graphs having exactly one cycle).

Cliques in edge intersection graphs of paths in a 1-tree. We have clearly edge-cliques, claw-cliques, cycle-cliques and, in addition, "pan-cliques", "bull-cliques", "net-cliques"(figure 2.9). In the best of our knowledge all maximal cliques are either edge-cliques, claw-cliques, cycle-cliques, pan-clique, bull-clique or net-cliques. That determining a maximum cardinality clique in a circular-arc graph is NP-hard implies also hardness for this case.

A lower bound for the span of any interval coloring of an edge intersection

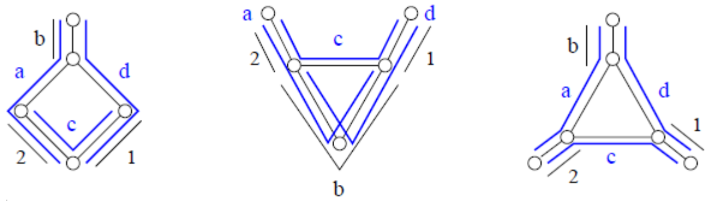

FIGURE 2.9: Pan-clique, Bull-clique and Net-clique resp. [11]

graph of paths in a 1-tree G is

$$\min\{\omega(I(\mathcal{P}), w) : \mathcal{P} \text{ possible routing in } G\}$$

which is clearly NP-hard to compute.

Superperfection of edge intersection graphs of paths in a 1-tree. If the network is a 1-tree, then all minimal non-superperfect graphs occuring in $I(\mathcal{P})$ when the network is a tree or a cycle can clearly be present. In addition, we can further show, by presenting according path collections, that also the families $\overline{E}_k$ and $\overline{F}_k$ can occur in such edge-intersection graphs. Thus, we obtain :

Theorem 2.4.1. *If $\mathcal{P}$ is a set of paths in a* 1-tree, *then $I(\mathcal{P})$ can contain $\overline{A}_1$, $\overline{A}_2$ and*

- *all odd holes C_{2k+1} and odd antiholes $\overline{C}_{2k+1}$ for $k \geq 2$,*
- *the graphs J_k for all $k \geq 2$ and J'_k for all $k \geq 3$,*
- *$\overline{D}_2$, $\overline{D}_3$, $\overline{D}_4$, but not the graphs $\overline{D}_k$ for $k \geq 5$,*
- *the graphs $\overline{E}_1$, $\overline{E}_2$, $\overline{E}_3$, $\overline{F}_1$, $\overline{F}_2$, $\overline{F}_3$, but none of $\overline{E}_k$, $\overline{F}_k$ for $k \geq 4$.*

Hence, we can finally conclude that *all* the studied minimal non-superperfect graphs can occur in edge-intersection graphs of paths, as soon as the network G is a 1-tree and satisfies minimal survivability conditions concerning edge or node failures.

Summary for edge intersection graphs of paths in 1-tree. We can summarize the relations of the lower bounds for the interval chromatic number as

$$\omega(I(\mathcal{P}), \mathbf{w}) \leq \chi(I(\mathcal{P}), \mathbf{w}) \leq \chi_I(I(\mathcal{P}), \mathbf{w}) \leq f(\omega(I(\mathcal{P}), \mathbf{w}))$$

and, thus, for the minimum spectrum width by

$$s_R(G, \mathcal{D}) \leq \omega(G, \mathcal{D}) \leq \chi(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D}) \leq \frac{3}{2}\omega(G, \mathcal{D}) \leq f(\omega(G, \mathcal{D})).$$

Problem 4.

- Are there other minimal non-superperfect intersection graphs different from the ones cited in Theorem 2.4.1 ?
- If there are examples of 1-trees with $s_R(G, \mathcal{D}) \leq \frac{2}{3}\omega(G, \mathcal{D})$ and other examples with $\omega(G, \mathcal{D}) \leq \frac{2}{3}\chi_I(G, \mathcal{D})$, are there cases with $s_R(G, \mathcal{D}) \leq \frac{4}{9}\chi_I(G, \mathcal{D})$ (some combination of the two cases below) ?

2.5 General Case

In the General case, for modern optical networks we can not assure that specific structures will describe the graph G. Actually, according to Kerivin and Wagler in [11], today's optical network are 2-connected, sparse planar graphs with small maximum degree and have more a grid-like structure. One example can be found in [23]. In this case, we have the follow theorem :

Theorem 2.5.1 (Kerivin and Wagler [11]). *All minimal non-comparability non-superperfect graphs can occur in edge intersection graphs I($\mathcal{P}$) of sets $\mathcal{P}$ of paths in optical networks G.*

Therefore, we can expect in the in edge intersection graphs I($\mathcal{P}$) and in some cases more than one one minimal non-comparability non-superperfect graphs. The possible presence of these structures brings the question how large can be the large between any two values in the follow chain :

$$s_R(G, \mathcal{D}) \leq \omega(G, \mathcal{D}) \leq \chi(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D})$$

3 - Proposal framework

3.1 General idea

The general idea of this framework is to be able to prove the optimality of the solution through the lower bound. So, we will build a solution to the routing problem without considering the assignment part yet and we will do it optimally. For this, we will use the mathematical model of min cost multi commodity flow, as will be explained in the section 3.2. With this answer, we will try to make the assignment in order to respect the solution found by the previous model. To build the assignment solution, we will use the edge-path formulation (that will be explained in the section 3.3) that selects a path for each demand and assigns slots to the demands.

In this framework we consider the structures that have already been presented, mainly the clicks to increase the lower bound whenever possible. The upper bound will be updated when the solution presented by the edge-path formulation is better than the current upper bound. In case the lower bound is equal to the upper bound we find the optimal solution.

We will also consider the cases that the instance is infeasible. in this case we will have to prove it either by making the upper bound larger than the lower bound, or because there are no routing options that respect the length restrictions.

3.2 Minimum cost multi commodity flow problem

To compute the $s_R(G, \mathcal{D})$, we can use the min-cost multi-commodity flow problem. We need to construct a new oriented graph G'(V',E') from the original graph G(V,E) in the RSA problem. Let V' be equal to V the set of vertices and E' = {l,l' : e ∈ E} the directed arcs where l and l' are in opposites directions and share the same vertices of e. We maintain the same set of demands $\mathcal{D}$.

Then, the objective is find the paths P_k to each demand $k \in \mathcal{D}$ that minimizes the flow passing by the arc who has the maximal flow in the RSA problem. We remark that the flow in each edge $e \in E$ will be equal to the sum of the flow in the pair l and l' arcs in the multi-commodity flow problem which share the same 2 vertices but are in opposites directions.

Let denote Cap the maximal sum of the flow passing by the pair of arcs l and l'. The others variables of the problem are

$$f_k(l) = \begin{cases} 1 & \text{if } l \in P_k \\ 0 & \text{otherwise} \end{cases}$$

This means that if the flow referent to the demand $k \in \mathcal{D}$ use the arc $l \in E'$ so the variable will have value 1. The objective function is to reduce the value of Cap. The first constraint, equation 3.1, assure that for each demand $k \in \mathcal{D}$ the flow will be sent from the origin vertex to the next vertex in P_k by exactly one arc. The second constraint, equation 3.2, assure that for each demand $k \in \mathcal{D}$ the flow will be sent to the destiny vertex from the last vertex in P_k by exactly one arc. The third constraint, equation 3.3, assure the flow preservation in all vertex in P_k for each $k \in \mathcal{D}$. There are also the length constraint 3.5 that assure that the path assigned to the demand k is not greater then the maximum allowed. The constraint 3.4 measures the maximal flow in one pair l and l'. Then, we have the domain variables constraint in the equation 3.6.

$$Minimize \ z = Cap$$

s.t.

$$\sum_{l \in \delta^-(o_k)} f_k(l) = 1 \qquad \forall \ k \in \mathcal{D} \tag{3.1}$$

$$\sum_{l \in \delta^+(d_k)} f_k(l) = 1 \qquad \forall \ k \in \mathcal{D} \tag{3.2}$$

$$\sum_{l \in \delta^-(v)} f_k(l) = \sum_{l \in \delta^+(v)} f_k(l) \qquad \forall \ k \in \mathcal{D} \ \ \forall \ v \neq o_k, d_k \tag{3.3}$$

$$\sum_{k \in \mathcal{D}} w_k f_k(l) + \sum_{k \in \mathcal{D}} w_k f_k(l') \ \leq \ Cap \qquad \forall \ pairs \ l, l' \in E' \tag{3.4}$$

$$\sum_{l \in E'} f_k(l) \bar{l}_l \ \leq \ \bar{l}_k \qquad \forall \ k \in \mathcal{D} \tag{3.5}$$

$$f_k(l) \in \{0, 1\} \qquad \forall \ k \in \mathcal{D} \ \forall \ arcs \ l \in E' \tag{3.6}$$

Therefore, the solution will indicate the paths for each demand such that the capacity of the all arcs are minimum and is possible to satisfy all demands with conflict-free. In that way, follows

$$Cap(G', \mathcal{D}) = s_R(G, \mathcal{D}) \leq \chi_I(G, \mathcal{D})$$

So, the solution of the min-cost multi-commodity flow problem give the solution to the *Routing Problem* represented by the variables $f_k(l)$ such that we have at most the minimal capacity obtained installed in all the arcs. This solution is a lower bound for the RSA problem represented by the variable Cap.

We remark that, in the case when the network is a path or a tree the unique routing possible will be found with this model. In the case of the networks is a path, holds

$$Cap(G', \mathcal{D}) = s_R(G, \mathcal{D}) = \omega(G, \mathcal{D}) = \chi(G, \mathcal{D})$$

and is not the optimal just in cases where the interval graph is not superperfect.

3.3 Edge-Path formulation

To compute the $\chi_I(G, \mathcal{D})$, we can use the Edge-Path Formulation. We need to construct a new problem from a set of considered paths $\bar{\mathcal{P}}$, that in this case will be proposed by the original minimum cost multi commodity flow problem. For constructing this problem, we use the set and the original graph G(V,E).

Then, the objective is find the spectrum assignment S and the chosen path P to each demand $k \in \mathcal{D}$ that minimizes the maximal number of used slots by the demands. This is done by assigning slots and choosing between one of the paths proposed for each demand . We remark that the used slots must be consecutive, so we add restrictions to respect this condition. We also remark that there's no limit to the number of proposed paths for each demand and this number does not need to be the same to all demands, as long as there is at least one option for each demand in the set $\bar{\mathcal{P}}$. From the graph we can also determine one upper bound $\bar{s}$ for the problem, namely the maximal edge load of the edges in E.

Let denote $\chi_I(G, \mathcal{D})$ the maximal number of slots used. The others variables of the problem are

$$y_{kp} = \begin{cases} 1 & \text{if } P_k \ \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases}$$

This means that, for each demand $k \in \mathcal{D}$, if the path p contained in the set P_k (contained in the set $\bar{\mathcal{P}}$) is chosen, the variable will have value 1.

$$z_{ks} \geq \ 0$$

This means that, for each demand $k \in \mathcal{D}$, z is the slot where the spectrum assignment ends. A demand with $z = 4$ and $w = 3$ would use the slots $[1, 4]$ for example.

$$x_{ke} = \begin{cases} 1 & \text{if } e \in P_k \\ 0 & \text{otherwise} \end{cases}$$

This means that, for each demand $k \in \mathcal{D}$, if the edge e is used by the demand, the variable will have value 1.

$$t_{kes} = \begin{cases} 1 & \text{if } s \in S_k, \ e \in P_k \\ 0 & \text{otherwise} \end{cases}$$

This means that, for each demand $k \in \mathcal{D}$, if the slot s is used in the edge e, the variable will have value 1. The objective function is to reduce the value of $\chi_I(G, \mathcal{D})$. The first constraint, equation 3.7, assure that for each demand $k \in \mathcal{D}$ one path will be chosen to attend this demand. The second constraint, equation

3.8, assure that for each demand $k \in \mathcal{D}$, if the path chosen use the edge e, so the variable x_{ke} will be assigned to 1 for all edges in the path. The third constraint, equation 3.9, assure the path chosen respect the length constraint for each demand $k \in \mathcal{D}$. The next constraint, equation 3.10 assign the slots for each demand and the assigned needs to be in one slot greater or equal to the weight of this demand, which is also assured by equation 3.11. The next constraint, equation 3.12, assign the variables t_{kes} whenever necessary by the other variables and is followed by equation 3.13 that assure that the number of slots assigned for each demand in each edge in the path chosen to this demand is exactly equal to the weight of this demand. The constraint represented by the equation 3.14 is a non-overlapping constraint , meaning that in one edge e and one slot s at most one demand can be assigned. The constraint 3.15 guarantee that the variable χ_I is greater or equal to the maximum slot assigned to all demands. Then, we have the domain variables constraints in the equations 3.16, 3.17, 3.18, 3.19 and 3.20.

$$Minimize\ z = \chi_I$$

s.t.

$$\sum_{P \in \bar{\mathcal{P}}_k} y_{kp} \ = \ 1 \qquad \forall\, k \in \mathcal{D} \tag{3.7}$$

$$\sum_{P \in \bar{\mathcal{P}}_k : e \in P} y_{kp} \ = \ x_{ke} \qquad \forall\, k \in \mathcal{D},\ e \in E \tag{3.8}$$

$$\sum_{e \in e} x_{ke} \bar{l}_e \ \leq \ \bar{l}_k \qquad \forall\, k \in \mathcal{D} \tag{3.9}$$

$$\sum_{s \in S : s >= w_k} z_{ks} \ = \ 1 \qquad \forall\, k \in \mathcal{D} \tag{3.10}$$

$$\sum_{s=0}^{w_k} z_{ks} = 0 \qquad \forall\, k \in \mathcal{D} \tag{3.11}$$

$$\sum_{i=0}^{w_k} z_{k,s+i} \ + x_{ke} \ <= \ t_{kes} + 1 \qquad \forall\, k \in \mathcal{D},\ e \in E, s \in S \tag{3.12}$$

$$\sum_{s \in S} t_{kes} \ = \ w_k x_{ke} \qquad \forall\, k \in \mathcal{D},\ e \in E \tag{3.13}$$

$$\sum_{k \in \mathcal{D}} t_{kes} \ \leq \ 1 \qquad \forall\, e \in E,\ s \in S \tag{3.14}$$

$$\sum_{s \in S}(s+1)z_{ks} \;\leq\; \chi_I \qquad \forall\, k \in \mathcal{D},\; s \in S \tag{3.15}$$

$$y_{kp} \in \{0,1\} \qquad \forall\, k \in \mathcal{D},\; \forall\, P \in \bar{\mathcal{P}} \tag{3.16}$$

$$x_{ke} \in \{0,1\} \qquad \forall\, k \in \mathcal{D},\; \forall\, e \in E \tag{3.17}$$

$$z_{ks} \in \{0,1\} \qquad \forall\, k \in \mathcal{D},\; \forall\, s \in S \tag{3.18}$$

$$t_{kes} \in \{0,1\} \qquad \forall\, k \in \mathcal{D},\; \forall\, e \in E, \forall\, s \in S \tag{3.19}$$

$$\chi_I \;\geq\; 0 \tag{3.20}$$

3.4 RSA framework

In this section we will present a framework for column generation for an edge-path formulation for routing and spectrum assignment in optical networks, driven by combinatorial properties, but only taking non-edge cliques into account, proposed by Kerivin and Wagler [11] :
Input : an instance $(G, \mathcal{D}, \bar{s})$
Output : a solution $(\mathcal{P}^*, \mathcal{S}^*)$ with span $\chi_I(G, \mathcal{D})$ or a certificate for infeasibility

1. Initialize
   — an upper bound by $b_{up} = \bar{s}$,
   — a set of considered paths by $\bar{\mathcal{P}} = \emptyset$,
   — a set of previously used routings by $\mathcal{R} = \emptyset$,
   — a set of non-edge cliques by $\mathcal{Q} = \emptyset$.

2. Compute a min-cost multi-commodity flow f in an auxiliary network G_f (constructed from G by replacing every edge $e = uv$ of G by a pair of arcs $a = uv, \bar{a} = vu$).
   If no feasible solution has been found then
   — return "instance infeasible (due to transmission reach constraints)"
   Else (set a lower bound) : let $b_{low} = s_R(G, \mathcal{D})$.

   If $b_{low} > \bar{s}$ then
   — return “instance infeasible (as $s_R(G, \mathcal{D}) > \bar{s}$)”

3. Determine from this flow f the according routing $\mathcal{P}_f$ and let $\bar{\mathcal{P}} = \bar{\mathcal{P}} \cup \mathcal{P}_f$. Launch the edge-path formulation with $\bar{\mathcal{P}}$ as set of paths enhanced by
   — forbidden clique constraints for all $Q \in \mathcal{Q}$
$$\sum_{P_k \in \mathcal{P}_Q} x_{P_k} \leq |Q| - 1$$
   — forbidden routing constraints for all $\mathcal{P} \in \mathcal{R}$
$$\sum_{P_k \in \mathcal{P}} x_{P_k} \leq |\mathcal{D}| - 1$$
   as minimum violation problem where the use of frequency slots
   — within $[1, b_{low}]$ does not cause any costs
   — within $[b_{low} + 1, b_{up}]$ is penalized
   and the objective is to minimize penalties.
   If no feasible solution has been found : let $\mathcal{R} = \mathcal{R} \cup \{\mathcal{P}_f\}$ and go to Step 5.

4. Evaluate solution $(\mathcal{P}, \mathcal{S})$ found :

   If $(\mathcal{P}, \mathcal{S})$ has objective function value $z^* = 0$ then
   — return $(\mathcal{P}, \mathcal{S})$ as optimal solution
   Else (we have $z^* > 0$ and try to improve b_{up}) :
   — let $\mathcal{R} = \mathcal{R} \cup \{\mathcal{P}_f, \mathcal{P}\}$ and identify max-used slot $s^* = s_{max}(\mathcal{S})$,
   — if $b_{up} > s^*$ then let $b_{up} = s^*$, keep $(\mathcal{P}, \mathcal{S})$ as currently best solution $(\mathcal{P}^*, \mathcal{S}^*)$
   If $(\mathcal{P}^*, \mathcal{S}^*)$ does not yet exist, keep $(\mathcal{P}, \mathcal{S})$ as currently best solution $(\mathcal{P}^*, \mathcal{S}^*)$

   Determine from $(\mathcal{P}, \mathcal{S})$ the subset $\mathcal{D}_c \subset \mathcal{D}$ of critical demands k whose channel $S_k \in \mathcal{S}$ uses frequency slots within $[b_{low} + 1, b_{up}]$.

   For each critical demand $k \in \mathcal{D}_c$:
   — construct the subgraph H_k of $I(\mathcal{P})$ induced by $N[k]$
   — find in H_k all cliques Q of weight $w(Q) > b_{low}$ and include them in $\mathcal{Q}$ as triple $(\mathcal{P}_Q, E_Q, w(Q))$ with
$$\mathcal{P}_Q = \{P_k \in \mathcal{P} : k \in Q\}$$
   and E_Q subset of edges of G where paths from $\mathcal{P}_Q$ meet.

5. Relaunch the multi-commodity flow as feasibility problem
   — with capacity $\leq b_{low}$ for all pairs of arcs and enhanced by
   — forbidden routing constraints associated with $\mathcal{P} \in \mathcal{R}$
$$\sum_{k \in \mathcal{D}} \sum_{a \in A^k_{\mathcal{P}}} f_k(a) \leq \sum_{k \in \mathcal{D}} |A^k_{\mathcal{P}}| - 1$$

where $A^k_{\mathcal{P}}$ denotes the subset of arcs with $f_k(a) > 0$ in $\mathcal{P}$
— forbidden clique constraints associated with $Q \in \mathcal{Q}$

$$\sum_{k \in Q} \sum_{a \in A^k_Q} f_k(a) \leq \sum_{k \in Q} |A^k_Q| - 1$$

with A^k_Q subset of arcs a corresponding to edges in E_Q with $f_k(a) > 0$.
If a feasible solution f has been found, then continue with Step 3.

6. Improvement of b_{low} :
If $\mathcal{Q} \neq \emptyset$ then
— let $b_{low} = \min\{w(Q) : Q \in \mathcal{Q}\}$,
— remove from $\mathcal{Q}$ all cliques Q of (new) weight $w(Q) = b_{low}$
Else (i.e. if $\mathcal{Q} = \emptyset$) let $b_{low} = b_{low} + 1$
(this corresponds to the special case that all critical configurations are non-superperfect graphs or no feasible solution has been found so far)

Test for termination :
If (we now have) $b_{low} > b_{up}$ then
— return "instance infeasible (as $\chi_I(G, \mathcal{D}) > \bar{s}$)"
(this corresponds to the case that no feasible solution has been found)
If (we now have) $b_{low} = b_{up}$ and $(\mathcal{P}^*, \mathcal{S}^*)$ exists then
— return $(\mathcal{P}^*, \mathcal{S}^*)$ as optimal solution
Else continue with Step 5 (with new b_{low})

4 - Implementation and tests

4.1 Implementation

The framework implementation was made using the programming lenguage C++. In addition to the standart C++ libraries the only other library used was Concert Technology, making possible to include the CPLEX solver, developed and offered by IBM.

To begin the implementation, we had one base code avaible, with some classes already created for the same problem. However, we made the choice to recreate the code from zero. This choice was motivated by many factors such as : the code available did not respect many principles of object programming, did not respect encapsulation rules and there were many attributes and methodes in the classes with unclear meaning and use. Other important factor was the amount of compiler warnings in each execution, derived from misuse of types, not used variables and library warnings.

Thus, the following classes were implemented : Vertex, Edge, Arc, Graph, Clique, Path, Demand, RSA Input, Output RSA, Output MCMCF and RSA Algorithms. The classes range from reading the instances following the model proposed by the examples to printing the final result found.

The code behaves as follows. First, the vertices of the network are created and then the edges are added. Soon after, the demands are created. This is all the information we have about the instances. and is done by an RSA_Input method.

This input will be the starting point for applications of the proposed algorithms. Some attributes are key in this class : the possible paths for each demand found, the routings found and the clicks that are forbidden, all initialized empty. Another important part of the initialization is the construction of the auxiliary graphe that will be used to solve the min cost multi commodity flow problem.

In the main function, calling the framework method is sufficient for the whole problem to be executed. Thus, after the object's initialization, the following steps will be performed according to the framework used as the basis foresees :

Our framework is implemented as follows : While the optimal solution (can be infeasible) is not yet found) :

1. Initialize
   - an lower bound by $b_{low} = 0$,
   - an upper bound by $b_{up} = \bar{s}$,
   - a set of considered paths by $\bar{\mathcal{P}} = \emptyset$,
   - a set of previously used routings found by the edge-path formulation by $\mathcal{R} = \emptyset$,

— a set of previously used routings found by the min-cost multi commodity flow by $\mathcal{R}_{mcmcf} = \emptyset$,
— a set of non-edge cliques by $\mathcal{Q} = \emptyset$.
— $iterator = 0$

2. Compute a min-cost multi-commodity flow f in an auxiliary network G_f (constructed from G by replacing every edge $e = uv$ of G by a pair of arcs $a = uv, \bar{a} = vu$).
   If no feasible solution has been found then
   — return "instance infeasible"
   Else (set a lower bound) : let $b_{low} = s_R(G, \mathcal{D})$.

   If $b_{low} > \bar{s}$ then
   — return "instance infeasible (as $s_R(G, \mathcal{D}) > \bar{s}$)"
3. Determine from this flow f the according routing $\mathcal{P}_f$ and let $\bar{\mathcal{P}} = \bar{\mathcal{P}} \cup \mathcal{P}_f$ and $\mathcal{R}_{mcmcf} = \mathcal{R}_{mcmcf} \cup \mathcal{P}_f$. Launch the edge-path formulation with $\bar{\mathcal{P}}$ as set of paths enhanced by
   — forbidden routing constraints for all $\mathcal{P} \in \mathcal{R}$

$$\sum_{P_k \in \mathcal{P}} x_{P_k} \leq |\mathcal{D}| - 1$$

   and objective function equal to χ_I If no feasible solution has been found : let $\mathcal{R} = \mathcal{R} \cup \{\mathcal{P}_f\}$ and go to Step 5.

4. Evaluate solution $(\mathcal{P}, \mathcal{S})$ found : If $(\mathcal{P}, \mathcal{S})$ has objective function value $z^* = b_{low}$ then
   — return $(\mathcal{P}, \mathcal{S})$ as optimal solution
   Else (we have $z^* > b_{low}$) :
   — let $\mathcal{R} = \mathcal{R} \cup \mathcal{P}$
   — construct the subgraph H_k of $I(\mathcal{P})$
   — find in H_k all cliques Q of weight $w(Q) > b_{low}$ and include them in $\mathcal{Q}$ as triple $(\mathcal{P}_Q, E_Q, w(Q))$ with

$$\mathcal{P}_Q = \{P_k \in \mathcal{P} : k \in Q\}$$

   and E_Q subset of edges of G where paths from $\mathcal{P}_Q$ meet.
5. Relaunch the multi-commodity flow problem enhanced by
   — forbidden routing constraints associated with $\mathcal{P} \in \mathcal{R}$

$$\sum_{k \in \mathcal{D}} \sum_{a \in A^k_{\mathcal{P}}} f_k(a) \leq \sum_{k \in \mathcal{D}} |A^k_{\mathcal{P}}| - 1$$

   where $A^k_{\mathcal{P}}$ denotes the subset of arcs with $f_k(a) > 0$ in $\mathcal{P}$

— forbidden routing constraints associated with $\mathcal{P} \in \mathcal{R}_{mcmcf}$

$$\sum_{k\in\mathcal{D}}\sum_{a\in A^k_{\mathcal{P}}} f_k(a) \leq \sum_{k\in\mathcal{D}} |A^k_{\mathcal{P}}| - 1$$

where $A^k_{\mathcal{P}}$ denotes the subset of arcs with $f_k(a) > 0$ in $\mathcal{P}$

— forbidden clique constraints associated with $Q \in \mathcal{Q}$

$$\sum_{k\in Q}\sum_{a\in A^k_{Q}} f_k(a) \leq \sum_{k\in Q} |A^k_{Q}| - 1$$

with A^k_Q subset of arcs a corresponding to edges in E_Q.

If a feasible solution f has been found,

— If $s_R(G, \mathcal{D}) = z^*max(\mathcal{P} \in \mathcal{R})$ then return $(\mathcal{P}, \mathcal{S})$ associated with P as Optimal Solution.

— Else if $s_R(G, \mathcal{D}) \geq b_{up}$ then return "instance infeasible"

— Else if $s_R(G, \mathcal{D}) \geq b_{low}$ then $b_{low} = s_R(G, \mathcal{D}$ and remove from $\mathcal{Q}$ all cliques Q of (new) weight $w(Q) \leq b_{low}$

— Let $\mathcal{R}_{mcmcf} \cup \mathcal{P}_f$ and let $\bar{\mathcal{P}} = \bar{\mathcal{P}} \cup \mathcal{P}_f$

— then continue with Step 3.

Else return "instance infeasible"

There are some differences in comparison with the proposal framework that we remark :

— We forbid in the min-cost multi commodity flow problems the routings previously found by the min-cost multi commodity flow problem and the routings previously found by the edge-path formulation. The explanation is that everytime we launch the edge-path formulation we consider all the paths found, so any combination of these paths never give to us a better result than one found by the edge-path formulation. So if there are no optimal solution yet, we can forbid all combination of paths already found.

— We don't launch feasibility problems. Instead of that, we choose to always launch an optimization problem and compare with the results already found.

— We don´t look for cliques just in the neighborhood of critical demands but in all demands. This is a difficulty that we found in the first try to identify the cliques and to simplify we made this choice. We think that this improvement could be done if we had more time and remark this as perspective of continuation of this project.

— The clique constraint does not consider only the arcs with flow equal to 1 in the routing but also the other arc in the pair that forms an edge. However, we maintain the same right side of the equation. It's a way to assure that if the routing intersect in this edges in any direction the clique needs to be yet forbidden.

Also, during the execution of tests we observed some necessary adjustments. We remark first our decision to forbid sub cycles that maybe will not make the solution of the min-cost multi commodity flow worst but can be a problem for us in the framework proposed. We first observe that in some solutions, CPLEX assigned the arcs a and $\bar{a}$ that represents the same edge for one demand. So we forbid this that element in the solution by the following constraint :

$$f_k(l) + f_k(l') \ \leq \ 1 \qquad \forall k \in \mathcal{D} \forall \ pairs \ l, l' \in E' \tag{4.1}$$

So, we observe that there were other cycles intersecting in the path of the assigned demand. Then, we created the following constraint to solve the problem :

$$\sum_{l \in \delta^-(v)} f_k(l) \ \leq 1 \qquad \forall \ k \in \mathcal{D} \ \ \forall \ v \neq o_k \tag{4.2}$$

Therefore, this two constraints, combined with a construction of the solution that just take into account the exact path from origin to destiny, we were able to find the right solution.

Another detail is that the instance has a upper bound specified but in the instances model each edge has its own upper bound. So, the trivial solution was to assign to all edges in the instance, the same upper bound of the problem.

4.2 Tests

The tests that were done are the first tests in relation to the implementation and the proposed framework. The main objective of these tests is to achieve the expected final result, thus validating the framework as well as the implementation made.

There are four instances for testing proposed for this project. It is important to note that the instances, despite being small, seek to test a specific part of the framework and implementation. instances can be described as :

Test instance 1

Given an optical network G represented by 4.1 (with all edges of length 1), an optical spectrum $S = \{1, \ldots, \bar{s}\}$ with $\bar{s} = 5$, and a set $\mathcal{D}$ of demands $k = (o_k, d_k, \bar{l}_k, w_k)$:

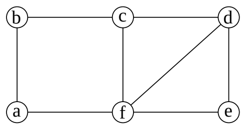

Figure 4.1: Test instance 1 graph.

k	$o_k \rightarrow d_k$	$\bar{l}_k$	w_k
1	$a \rightarrow c$	4	2
2	$a \rightarrow d$	4	1
3	$b \rightarrow f$	4	2
4	$b \rightarrow e$	4	1
5	$d \rightarrow f$	4	3

The solution found by the solver is :

===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 5 - 4 - 3 : : with length 4
Slots : 1 - 3
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3

Slots : 0 - 1
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 6 : : with length 2
Slots : 1 - 3
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
Slots : 3 - 4
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
Slots : 0 - 3

The steps until finding the solution are better described in the appendix A. We believe that the solver work correctly and found the right solution.

Test instance 2

Given an optical network G represented by 4.2 (with indicated edge lengths), an optical spectrum $S = \{1, \ldots, \bar{s}\}$ with $\bar{s} = 8$, and a set $\mathcal{D}$ of demands $k = (o_k, d_k, \bar{l}_k, w_k)$:

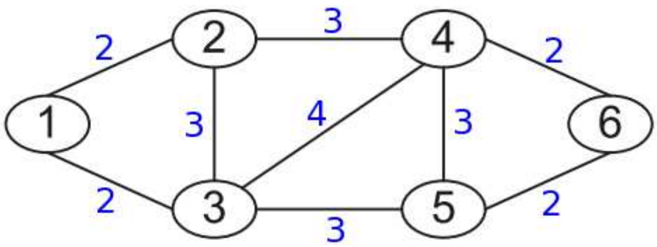

FIGURE 4.2: Test instance 2 graph.

k	$o_k \to d_k$	$\bar{l}_k$	w_k
1	$1 \to 6$	$\bar{l}_1$	3
2	$1 \to 5$	$\bar{l}_2$	1
3	$2 \to 5$	$\bar{l}_3$	3
4	$2 \to 6$	$\bar{l}_4$	1
5	$3 \to 6$	$\bar{l}_5$	3
6	$4 \to 1$	$\bar{l}_6$	2

Two different runs are proposed by this instance. The first one is with $\bar{l}_k = 6$ for all k,

The solution found by the solver is :

The first iteration of MCMCF is impossible for this instance
Problem is infeasible

The steps until finding the solution are better described in the appendix B. We believe that the solver work correctly and found the right solution.

The second one is with $\bar{l}_k = 7$ for all k. The solution found by the solver is :

===== SOLUTION =====
This RSA solution matches the lowerBound_
Output - RSA Problem -
Interval chromatic number = 6
Routing :
From demand : 1 Between nodes : 1 to 6 : : 1 - 3 - 5 - 6 : : with length 7
Slots : 1 - 4
From demand : 2 Between nodes : 1 to 5 : : 1 - 3 - 5 : : with length 5
Slots : 0 - 1
From demand : 3 Between nodes : 2 to 5 : : 2 - 4 - 5 : : with length 6
Slots : 3 - 6
From demand : 4 Between nodes : 2 to 6 : : 2 - 4 - 6 : : with length 5
Slots : 0 - 1
From demand : 5 Between nodes : 3 to 6 : : 3 - 4 - 6 : : with length 6
Slots : 2 - 5
From demand : 6 Between nodes : 4 to 1 : : 4 - 2 - 1 : : with length 5
Slots : 1 - 3

The steps until finding the solution are better described in the appendix C. We believe that the solver work correctly and found the right solution.

Test instance 3

Given an optical network G represented by 4.3 (with all edges of length 1), an optical spectrum $S = \{1, \ldots, \bar{s}\}$ with $\bar{s} = 6$, and a set $\mathcal{D}$ of demands $k = (o_k, d_k, \bar{l}_k, w_k)$:

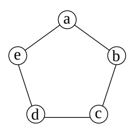

FIGURE 4.3: Test instance 3 graph.

k	$o_k \to d_k$	$\bar{l}_k$	w_k
1	$a \to c$	3	2
2	$b \to d$	3	2
3	$c \to e$	3	2
4	$d \to a$	3	2
5	$e \to b$	3	2

The solution found by the solver is :

===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped
Output - RSA Problem -
Interval chromatic number = 6
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 2 - 3 : : with length 2
Slots : 2 - 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 3 - 4 : : with length 2
Slots : 4 - 6
From demand : 3 Between nodes : 3 to 5 : : 3 - 4 - 5 : : with length 2
Slots : 2 - 4
From demand : 4 Between nodes : 4 to 1 : : 4 - 5 - 1 : : with length 2
Slots : 4 - 6
From demand : 5 Between nodes : 5 to 2 : : 5 - 1 - 2 : : with length 2
Slots : 0 - 2

The steps until finding the solution are better described in the appendix D. We believe that the solver work correctly and found the right solution.

Test instance 4

Given an optical network G represented by 4.4 (with all edges of length 1), an optical spectrum $S = \{1, \ldots, \bar{s}\}$ with $\bar{s} = 5$, and a set $\mathcal{D}$ of demands $k = (o_k, d_k, \bar{l}_k, w_k)$:

k	$o_k \to d_k$	$\bar{l}_k$	w_k
1	$a \to c$	4	1
2	$b \to d$	4	1
3	$c \to a$	4	1
4	$d \to b$	4	1

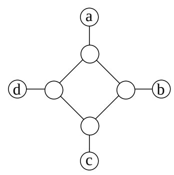

FIGURE 4.4: Test instance 4 graph.

The solution found by the solver is :

===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
Slots : 0 - 1
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 5 - 8 - 4 : : with length 4
Slots : 1 - 2
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
Slots : 3 - 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
Slots : 2 - 3

The steps until finding the solution are better better described in the appendix E. We believe that the solver work correctly and found the right solution.

Conclusion

This study aims to study the lower bound for the width of the Routing and Spectrum Assignment problem. For this, we first understand the problem and the lower bound chain proposed for this problem. Then, we study the characteristics of the lower bounds chain and the theoretical properties that are contained in them. Then we present the problem solving framework based on the properties that we can conclude from the theoretical study to finally implement the framework and present the results in the first tests.

The report describes that the work was completed according to the project's expectations. As difficulties encountered, we can mention some difficulties at the begining in assimilating graph structure concepts and their relationship with the solution that could be proposed. Later in the project there were some difficulties imposed by the confinement measures due to the pandemic caused by the Covid-19.

Finally, we believe that this project was important for our learning, notably regarding graph structures and their properties and the study of bounds and how we can propose solutions through them. Anyway, we hope that our contribution to the project is important and that our code and report can be useful for others to continue the project.

Perspectives

We believe that this work has many perspectives for continuation if we consider the theoretical properties presented and the framework implemented. Firstly, it is possible to improve the detection of cliques considering only the neighborhood of critical demands. Second, the framework presented considers only edge cliques, even though we can consider other types of cliques that are really quick to detect. We do not consider for example the detection of odd holes or odd anti-holes when comparing the lower bound with the result found. Finally, the implementation leaves much to be desired in relation to the theoretical opportunities presented by the report.

Regarding the theoretical part, the report is very clear when presenting the problems open or that we are not aware of the solution that can be useful in improving the established lower bound chain. It is important to note that many results presented in the state of the art are not considered, for example, at no time did we identify the structure of the graphs we are working on in our implementation. Thus, even though it has been extremely important for understanding, the properties shown in the state of the art have been not so much use full in the practice and can be a simple way to continue

Finally, for the mathematical models used, this work does not present a theoretical study on the complexity and does not do it. It is important to note that the presented framework can be extremely influenced by these results, so it is essential that they are presented if the option made is to continue with this framework for tests with larger instances.

References

[1] S. Talebi et al. “Spectrum management techniques for elastic optical networks : A survey”. In : *Optical Switching and Networking* 13 (2014), p. 34-48.

[2] Christodoulopoulos K., I. Tomkos et E. Varvarigos. “Elastic bandwidth allocation in flexible OFDM based optical networks”. In : *IEEE J. Lightwave Technol* 29 (2011), p. 1354-1366.

[3] Y. Wang, X. Cao et Y. Pan. “A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks”. In : *Proceedings of IEEE INFOCOM* (2011).

[4] S. Shirazipourazad et al. “On routing and spectrum allocation in spectrum sliced optical networks”. In : *Proceedings of IEEE INFOCOM* (2013), p. 385-389.

[5] Gröschel M.and Lovàsz L. et A. Schrijver. “Geometric Algorithms and Combinatorial Optimization”. In : *Springer-Verlag* (1988).

[6] Chudnovsky M. et al. “The Strong Perfect Graph Theorem”. In : *Annals of Mathematics* 164 (2006), p. 51-229.

[7] A. Hoffman. “A generalization of max flow-min cut”. In : *Math. Programming* 6 (1974), p. 352-359.

[8] M. Golumbic. “Algorithmic Graph Theory and Perfect Graphs”. In : *North Holland* 2nd Ed. (2004).

[9] T. Gallai. “Transitiv orientierbare Graphen”. In : *Acta Math. Acad. Sci. Hungar.* 18 (1967), p. 25-66.

[10] T. Andreae. “On superperfect noncomparability graphs”. In : *J. of Graph Theory* 9 (1985), p. 523-532.

[11] H. Kerivin et A. Wagler. “On superperfection of edge intersection graphs of paths”. In : *manuscript* ().

[12] F. Gavril. “Algorithms on circular-arc graphs”. In : *Networks* 4 (1974), p. 357-369.

[13] R. Jamison Golumbic M. “The Edge Intersection Graphs of Paths in a Tree”. In : *J. Comb. Theory B* 38 (1985), p. 8-22.

[14] M. Golumbic et M. Lipshteynand M. Stern. “Edge intersection graphs of single bend paths in a grid”. In : *Networks* 54 (2009), p. 130-138.

[15] A. Gyárfás. “Problems from the world surrounding perfect graphs”. In : *Zastos. Mat.* 19 (1987), p. 413-431.

[16] C Berge. "Les problemes de coloration en théorie des graphes". In : *Publl. Inst. Stat. Univ. Paris 9* (1960), p. 123-160.

[17] C. Lekkerkerker et D. Boland. "Representation of finite graphs by a set of intervals on the real line". In : *Fundamenta Mathematica* 51 (1962), p. 45-64.

[18] G. Durán, L.N. Grippo et M.D. Safe. "Structural results on circular-arc graphs and circle graphs : A survey and the main open problems". In : *Discrete Applied Mathematics* 164 (2014), p. 427-443.

[19] F. Gavril. "Algorithms for a maximum clique and a maximum independent set of a circle graph". In : *Networks* 3 (1973), p. 261-273.

[20] J. Moore Trotter W. "Characterization problems for graphs, partially ordered sets, lattices, and families of sets". In : *Optical Switching and Networking* 16 (1976), p. 361-381.

[21] F. Bonomo, G. Duránd L. Grippo et M. Safe. "Partial characterizations of circular-arc graphs". In : *Journal of Graph Theory* 61 (2009), p. 289-306.

[22] H. Kerivin et A.R. Mahjoub. "Design of survivable networks : A survey". In : *Networks* 46 (2005), p. 1-21.

[23] M. Ruiz et al. "Column generation algorithm for RSA problems in flexgrid optical networks". In : *Photonic Network Communications* 26 (2013), p. 53-64.

Table of annexes

A - Solution of Instance

FRAMEWORK FOR COLUMN GENERATION FOR AN EDGE PATH FORMULATION
==================BEGIN READING DATA===============
==================END READING DATA===============
——————BEGIN OF Data Information———————
Graph G
Number of nodes : 6| Number of links : : 8
Nodes : (1 2 3 4 5 6)
Not oriented
Edges : (1,2) Length 1
(1,6) Length 1
(2,3) Length 1
(3,4) Length 1
(3,6) Length 1
(4,5) Length 1
(4,6) Length 1
(5,6) Length 1
———
Graph G prime
Number of nodes : 6| Number of links : : 16
Nodes : (1 2 3 4 5 6)
Oriented
Arcs : (1,2) Length 1
(1,6) Length 1
(2,3) Length 1
(3,4) Length 1
(3,6) Length 1
(4,5) Length 1
(4,6) Length 1
(5,6) Length 1
(2,1) Length 1
(6,1) Length 1
(3,2) Length 1
(4,3) Length 1
(6,3) Length 1
(5,4) Length 1
(6,4) Length 1

(6,5) Length 1
———
Demands
———Index : 1
Origin : 1| Destination : 3
Slots : 2| Max Lenght : 4
———Index : 2
Origin : 1| Destination : 4
Slots : 1| Max Lenght : 4
———Index : 3
Origin : 2| Destination : 6
Slots : 2| Max Lenght : 4
———Index : 4
Origin : 2| Destination : 5
Slots : 1| Max Lenght : 4
———Index : 5
Origin : 4| Destination : 6
Slots : 3| Max Lenght : 4
——————-END OF Data Information——————-
——————BEGIN OF ILP Min Cost Multi Commodity Cplex——————
Number of demands : 5
Number of arcs : 16
Number of nodes : 6
Number of variables = 81
Capacity :3
——————END OF ILP Min Cost Multi Commodity Cplex——————
Output - MCMCF Problem -
Capacity = 3
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 5 - 4 - 3 : : with length 4
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 6 : : with length 2
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
——————BEGIN OF Possible paths Information——————
For demand : 1
Edges : (1,6) (5,6) (4,5) (3,4) Path Lenght : 4
********************** :
For demand : 2
Edges : (1,2) (2,3) (3,4) Path Lenght : 3

*********************** :
For demand : 3
Edges : (2,3) (3,6) Path Lenght : 2
*********************** :
For demand : 4
Edges : (1,2) (1,6) (5,6) Path Lenght : 3
*********************** :
For demand : 5
Edges : (4,6) Path Lenght : 1
*********************** :
——————END OF Possible paths Information———————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 5
Number of Edges : 8
Number of nodes : 6
Number of slices from MCMCF : 5
Number of variables = 271
Objective Value : 4
——————END OF ILP Routing And Spectrum Assignment Cplex———————
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 5 - 4 - 3 : : with length 4
Slots : 1 - 3
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3
Slots : 0 - 1
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 6 : : with length 2
Slots : 1 - 3
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
Slots : 3 - 4
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
Slots : 0 - 3
——————BEGIN OF Forbidden Cliques Information———————
Size : 3| Omega : 4
Clique : (1 2 4)
———
——————END OF Forbidden Cliques Information———————
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 5

Number of arcs : 16
Number of nodes : 6
Number of variables = 81
Capacity :3
——————END OF ILP Min Cost Multi Commodity Cplex———————
Output - MCMCF Problem -
Capacity = 3
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 3 : : with length 2
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 4 - 5 - 6 : : with length 4
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
——————BEGIN OF Possible paths Information———————
For demand : 1
Edges : (1,6) (5,6) (4,5) (3,4) Path Lenght : 4
Edges : (1,6) (3,6) Path Lenght : 2
*********************** :
For demand : 2
Edges : (1,2) (2,3) (3,4) Path Lenght : 3
*********************** :
For demand : 3
Edges : (2,3) (3,6) Path Lenght : 2
Edges : (2,3) (3,4) (4,5) (5,6) Path Lenght : 4
*********************** :
For demand : 4
Edges : (1,2) (1,6) (5,6) Path Lenght : 3
*********************** :
For demand : 5
Edges : (4,6) Path Lenght : 1
*********************** :
——————END OF Possible paths Information———————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 5
Number of Edges : 8
Number of nodes : 6
Number of slices from MCMCF : 5
Number of variables = 273
Objective Value : 4

——————END OF ILP Routing And Spectrum Assignment Cplex——————
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 3 : : with length 2
Slots : 0 - 2
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3
Slots : 0 - 1
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 4 - 5 - 6 : : with length 4
Slots : 1 - 3
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
Slots : 3 - 4
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
Slots : 0 - 3
——————BEGIN OF Forbidden Cliques Information——————
Size : 3| Omega : 4
Clique : (1 2 4)
———
Size : 3| Omega : 4
Clique : (2 3 4)
———
——————END OF Forbidden Cliques Information——————
——————BEGIN OF ILP Min Cost Multi Commodity Cplex——————
Number of demands : 5
Number of arcs : 16
Number of nodes : 6
Number of variables = 81
Capacity :4
——————END OF ILP Min Cost Multi Commodity Cplex——————
Output - MCMCF Problem -
Capacity = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 2 - 3 : : with length 2
From demand : 2 Between nodes : 1 to 4 : : 1 - 6 - 4 : : with length 2
From demand : 3 Between nodes : 2 to 6 : : 2 - 1 - 6 : : with length 2
From demand : 4 Between nodes : 2 to 5 : : 2 - 3 - 4 - 5 : : with length 3
From demand : 5 Between nodes : 4 to 6 : : 4 - 3 - 6 : : with length 2
===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped
Output - RSA Problem -

Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 6 - 5 - 4 - 3 : : with length 4
Slots : 1 - 3
From demand : 2 Between nodes : 1 to 4 : : 1 - 2 - 3 - 4 : : with length 3
Slots : 0 - 1
From demand : 3 Between nodes : 2 to 6 : : 2 - 3 - 6 : : with length 2
Slots : 1 - 3
From demand : 4 Between nodes : 2 to 5 : : 2 - 1 - 6 - 5 : : with length 3
Slots : 3 - 4
From demand : 5 Between nodes : 4 to 6 : : 4 - 6 : : with length 1
Slots : 0 - 3
———————————————— :
Iterations : 3
—————-END OF EXECUTION——————- :

B - Solution of Instance 2.1

FRAMEWORK FOR COLUMN GENERATION FOR AN EDGE PATH FORMULATION
===================BEGIN READING DATA===============
===================END READING DATA===============
————BEGIN OF Data Information————
Graph G
Number of nodes : 6| Number of links : : 9
Nodes : (1 2 3 4 5 6)
Not oriented
Edges : (1,2) Length 2
(1,3) Length 2
(2,3) Length 3
(2,4) Length 3
(3,4) Length 4
(3,5) Length 3
(4,5) Length 3
(4,6) Length 2
(5,6) Length 2
——
Graph G prime
Number of nodes : 6| Number of links : : 18
Nodes : (1 2 3 4 5 6)
Oriented
Arcs : (1,2) Length 2
(1,3) Length 2
(2,3) Length 3
(2,4) Length 3
(3,4) Length 4
(3,5) Length 3
(4,5) Length 3
(4,6) Length 2
(5,6) Length 2
(2,1) Length 2
(3,1) Length 2
(3,2) Length 3
(4,2) Length 3
(4,3) Length 4

(5,3) Length 3
(5,4) Length 3
(6,4) Length 2
(6,5) Length 2
———
Demands
———Index : 1
Origin : 1| Destination : 6
Slots : 3| Max Lenght : 6
———Index : 2
Origin : 1| Destination : 5
Slots : 1| Max Lenght : 6
———Index : 3
Origin : 2| Destination : 5
Slots : 3| Max Lenght : 6
———Index : 4
Origin : 2| Destination : 6
Slots : 1| Max Lenght : 6
———Index : 5
Origin : 3| Destination : 6
Slots : 3| Max Lenght : 6
———Index : 6
Origin : 4| Destination : 1
Slots : 2| Max Lenght : 6
——————-END OF Data Information———————-
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 6
Number of arcs : 18
Number of nodes : 6
Number of variables = 109
The first iteration of MCMCF is impossible for this instance
Problem is infeasible
———————————————————— :
Iterations : 1
——————-END OF EXECUTION———————- :

C - Solution of Instance 2.2

FRAMEWORK FOR COLUMN GENERATION FOR AN EDGE PATH FORMULATION
==================BEGIN READING DATA===============
==================END READING DATA===============
——————BEGIN OF Data Information———————
Graph G
Number of nodes : 6| Number of links : : 9
Nodes : (1 2 3 4 5 6)
Not oriented
Edges : (1,2) Length 2
(1,3) Length 2
(2,3) Length 3
(2,4) Length 3
(3,4) Length 4
(3,5) Length 3
(4,5) Length 3
(4,6) Length 2
(5,6) Length 2
———
Graph G prime
Number of nodes : 6| Number of links : : 18
Nodes : (1 2 3 4 5 6)
Oriented
Arcs : (1,2) Length 2
(1,3) Length 2
(2,3) Length 3
(2,4) Length 3
(3,4) Length 4
(3,5) Length 3
(4,5) Length 3
(4,6) Length 2
(5,6) Length 2
(2,1) Length 2
(3,1) Length 2
(3,2) Length 3
(4,2) Length 3
(4,3) Length 4

(5,3) Length 3
(5,4) Length 3
(6,4) Length 2
(6,5) Length 2
——
Demands
——Index : 1
Origin : 1| Destination : 6
Slots : 3| Max Lenght : 7
——Index : 2
Origin : 1| Destination : 5
Slots : 1| Max Lenght : 7
——Index : 3
Origin : 2| Destination : 5
Slots : 3| Max Lenght : 7
——Index : 4
Origin : 2| Destination : 6
Slots : 1| Max Lenght : 7
——Index : 5
Origin : 3| Destination : 6
Slots : 3| Max Lenght : 7
——Index : 6
Origin : 4| Destination : 1
Slots : 2| Max Lenght : 7
————-END OF Data Information—————-
————BEGIN OF ILP Min Cost Multi Commodity Cplex—————
Number of demands : 6
Number of arcs : 18
Number of nodes : 6
Number of variables = 109
Capacity :6
————END OF ILP Min Cost Multi Commodity Cplex—————
Output - MCMCF Problem -
Capacity = 6
Routing :
From demand : 1 Between nodes : 1 to 6 : : 1 - 3 - 5 - 6 : : with length 7
From demand : 2 Between nodes : 1 to 5 : : 1 - 3 - 5 : : with length 5
From demand : 3 Between nodes : 2 to 5 : : 2 - 4 - 5 : : with length 6
From demand : 4 Between nodes : 2 to 6 : : 2 - 4 - 6 : : with length 5
From demand : 5 Between nodes : 3 to 6 : : 3 - 4 - 6 : : with length 6

From demand : 6 Between nodes : 4 to 1 : : 4 - 2 - 1 : : with length 5
——————BEGIN OF Possible paths Information———————
For demand : 1
Edges : (1,3) (3,5) (5,6) Path Lenght : 7
********************** :
For demand : 2
Edges : (1,3) (3,5) Path Lenght : 5
********************** :
For demand : 3
Edges : (2,4) (4,5) Path Lenght : 6
********************** :
For demand : 4
Edges : (2,4) (4,6) Path Lenght : 5
********************** :
For demand : 5
Edges : (3,4) (4,6) Path Lenght : 6
********************** :
For demand : 6
Edges : (2,4) (1,2) Path Lenght : 5
********************** :
——————END OF Possible paths Information———————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 6
Number of Edges : 9
Number of nodes : 6
Number of slices from MCMCF : 8
Number of variables = 541
Objective Value : 6
——————END OF ILP Routing And Spectrum Assignment Cplex———————
===== SOLUTION =====
This RSA solution matches the lowerBound
Output - RSA Problem -
Interval chromatic number = 6
Routing :
From demand : 1 Between nodes : 1 to 6 : : 1 - 3 - 5 - 6 : : with length 7
Slots : 1 - 4
From demand : 2 Between nodes : 1 to 5 : : 1 - 3 - 5 : : with length 5
Slots : 0 - 1
From demand : 3 Between nodes : 2 to 5 : : 2 - 4 - 5 : : with length 6

Slots : 3 - 6
From demand : 4 Between nodes : 2 to 6 : : 2 - 4 - 6 : : with length 5
Slots : 0 - 1
From demand : 5 Between nodes : 3 to 6 : : 3 - 4 - 6 : : with length 6
Slots : 2 - 5
From demand : 6 Between nodes : 4 to 1 : : 4 - 2 - 1 : : with length 5
Slots : 1 - 3
———————————————— :
Iterations : 1
—————-END OF EXECUTION——————- :

D - Solution of Instance 3

FRAMEWORK FOR COLUMN GENERATION FOR AN EDGE PATH FORMULATION
==================BEGIN READING DATA===============
==================END READING DATA===============
——————BEGIN OF Data Information———————
Graph G
Number of nodes : 5| Number of links : : 5
Nodes : (1 2 3 4 5)
Not oriented
Edges : (1,2) Length 1
(1,5) Length 1
(2,3) Length 1
(3,4) Length 1
(4,5) Length 1
———
Graph G prime
Number of nodes : 5| Number of links : : 10
Nodes : (1 2 3 4 5)
Oriented
Arcs : (1,2) Length 1
(1,5) Length 1
(2,3) Length 1
(3,4) Length 1
(4,5) Length 1
(2,1) Length 1
(5,1) Length 1
(3,2) Length 1
(4,3) Length 1
(5,4) Length 1
———
Demands
———Index : 1
Origin : 1| Destination : 3
Slots : 2| Max Lenght : 3
———Index : 2
Origin : 2| Destination : 4
Slots : 2| Max Lenght : 3

———Index : 3
Origin : 3| Destination : 5
Slots : 2| Max Lenght : 3
———Index : 4
Origin : 4| Destination : 1
Slots : 2| Max Lenght : 3
———Index : 5
Origin : 5| Destination : 2
Slots : 2| Max Lenght : 3
——————-END OF Data Information———————-
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 5
Number of arcs : 10
Number of nodes : 5
Number of variables = 51
Capacity :4
——————END OF ILP Min Cost Multi Commodity Cplex———————
Output - MCMCF Problem -
Capacity = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 2 - 3 : : with length 2
From demand : 2 Between nodes : 2 to 4 : : 2 - 3 - 4 : : with length 2
From demand : 3 Between nodes : 3 to 5 : : 3 - 4 - 5 : : with length 2
From demand : 4 Between nodes : 4 to 1 : : 4 - 5 - 1 : : with length 2
From demand : 5 Between nodes : 5 to 2 : : 5 - 1 - 2 : : with length 2
——————BEGIN OF Possible paths Information———————
For demand : 1
Edges : (1,2) (2,3) Path Lenght : 2
********************** :
For demand : 2
Edges : (2,3) (3,4) Path Lenght : 2
********************** :
For demand : 3
Edges : (3,4) (4,5) Path Lenght : 2
********************** :
For demand : 4
Edges : (4,5) (1,5) Path Lenght : 2
********************** :
For demand : 5
Edges : (1,5) (1,2) Path Lenght : 2

********************** :
—————END OF Possible paths Information——————
—————BEGIN OF ILP Routing And Spectrum Assignment Cplex—————
—
Number of demands : 5
Number of Edges : 5
Number of nodes : 5
Number of slices from MCMCF : 6
Number of variables = 211
Objective Value : 6
—————END OF ILP Routing And Spectrum Assignment Cplex——————
Output - RSA Problem -
Interval chromatic number = 6
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 2 - 3 : : with length 2
Slots : 2 - 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 3 - 4 : : with length 2
Slots : 4 - 6
From demand : 3 Between nodes : 3 to 5 : : 3 - 4 - 5 : : with length 2
Slots : 2 - 4
From demand : 4 Between nodes : 4 to 1 : : 4 - 5 - 1 : : with length 2
Slots : 4 - 6
From demand : 5 Between nodes : 5 to 2 : : 5 - 1 - 2 : : with length 2
Slots : 0 - 2
—————BEGIN OF Forbidden Cliques Information——————
—————END OF Forbidden Cliques Information——————
—————BEGIN OF ILP Min Cost Multi Commodity Cplex——————
Number of demands : 5
Number of arcs : 10
Number of nodes : 5
Number of variables = 51
Capacity :6
—————END OF ILP Min Cost Multi Commodity Cplex——————
Output - MCMCF Problem -
Capacity = 6
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 4 - 3 : : with length 3
From demand : 2 Between nodes : 2 to 4 : : 2 - 3 - 4 : : with length 2
From demand : 3 Between nodes : 3 to 5 : : 3 - 4 - 5 : : with length 2
From demand : 4 Between nodes : 4 to 1 : : 4 - 5 - 1 : : with length 2

From demand : 5 Between nodes : 5 to 2 : : 5 - 1 - 2 : : with length 2
===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped
Output - RSA Problem -
Interval chromatic number = 6
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 2 - 3 : : with length 2
Slots : 2 - 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 3 - 4 : : with length 2
Slots : 4 - 6
From demand : 3 Between nodes : 3 to 5 : : 3 - 4 - 5 : : with length 2
Slots : 2 - 4
From demand : 4 Between nodes : 4 to 1 : : 4 - 5 - 1 : : with length 2
Slots : 4 - 6
From demand : 5 Between nodes : 5 to 2 : : 5 - 1 - 2 : : with length 2
Slots : 0 - 2
———————————————— :
Iterations : 2
—————-END OF EXECUTION——————- :

E - Solution of Instance 4

FRAMEWORK FOR COLUMN GENERATION FOR AN EDGE PATH FORMULATION
===================BEGIN READING DATA===============
===================END READING DATA===============
——————BEGIN OF Data Information————————
Graph G
Number of nodes : 8| Number of links : : 8
Nodes : (1 2 3 4 5 6 7 8)
Not oriented
Edges : (1,5) Length 1
(2,6) Length 1
(3,7) Length 1
(4,8) Length 1
(5,6) Length 1
(5,8) Length 1
(6,7) Length 1
(7,8) Length 1
———
Graph G prime
Number of nodes : 8| Number of links : : 16
Nodes : (1 2 3 4 5 6 7 8)
Oriented
Arcs : (1,5) Length 1
(2,6) Length 1
(3,7) Length 1
(4,8) Length 1
(5,6) Length 1
(5,8) Length 1
(6,7) Length 1
(7,8) Length 1
(5,1) Length 1
(6,2) Length 1
(7,3) Length 1
(8,4) Length 1
(6,5) Length 1
(8,5) Length 1
(7,6) Length 1

(8,7) Length 1
———
Demands
———Index : 1
Origin : 1| Destination : 3
Slots : 1| Max Lenght : 4
———Index : 2
Origin : 2| Destination : 4
Slots : 1| Max Lenght : 4
———Index : 3
Origin : 3| Destination : 1
Slots : 1| Max Lenght : 4
———Index : 4
Origin : 4| Destination : 2
Slots : 1| Max Lenght : 4
——————-END OF Data Information———————-
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 4
Number of arcs : 16
Number of nodes : 8
Number of variables = 65
Capacity :2
——————END OF ILP Min Cost Multi Commodity Cplex———————
Output - MCMCF Problem -
Capacity = 2
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 5 - 8 - 4 : : with length 4
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
——————BEGIN OF Possible paths Information———————
For demand : 1
Edges : (1,5) (5,8) (7,8) (3,7) Path Lenght : 4
*********************** :
For demand : 2
Edges : (2,6) (5,6) (5,8) (4,8) Path Lenght : 4
*********************** :
For demand : 3
Edges : (3,7) (6,7) (5,6) (1,5) Path Lenght : 4
*********************** :

For demand : 4
Edges : (4,8) (7,8) (6,7) (2,6) Path Lenght : 4
********************** :
——————END OF Possible paths Information———————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 4
Number of Edges : 8
Number of nodes : 8
Number of slices from MCMCF : 5
Number of variables = 217
Objective Value : 4
——————END OF ILP Routing And Spectrum Assignment Cplex———————
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
Slots : 0 - 1
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 5 - 8 - 4 : : with length 4
Slots : 1 - 2
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
Slots : 3 - 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
Slots : 2 - 3
——————BEGIN OF Forbidden Cliques Information———————
Size : 4| Omega : 4
Clique : (1 2 3 4)
———
Size : 3| Omega : 3
Clique : (1 2 3)
———
Size : 3| Omega : 3
Clique : (1 2 4)
———
Size : 3| Omega : 3
Clique : (1 3 4)
———
Size : 3| Omega : 3
Clique : (2 3 4)
———

——————END OF Forbidden Cliques Information———————
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 4
Number of arcs : 16
Number of nodes : 8
Number of variables = 65
Capacity :2
——————END OF ILP Min Cost Multi Commodity Cplex———————
Output - MCMCF Problem -
Capacity = 2
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 6 - 7 - 3 : : with length 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 7 - 8 - 4 : : with length 4
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 8 - 5 - 1 : : with length 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 5 - 6 - 2 : : with length 4
——————BEGIN OF Possible paths Information———————
For demand : 1
Edges : (1,5) (5,8) (7,8) (3,7) Path Lenght : 4
Edges : (1,5) (5,6) (6,7) (3,7) Path Lenght : 4
********************** :
For demand : 2
Edges : (2,6) (5,6) (5,8) (4,8) Path Lenght : 4
Edges : (2,6) (6,7) (7,8) (4,8) Path Lenght : 4
********************** :
For demand : 3
Edges : (3,7) (6,7) (5,6) (1,5) Path Lenght : 4
Edges : (3,7) (7,8) (5,8) (1,5) Path Lenght : 4
********************** :
For demand : 4
Edges : (4,8) (7,8) (6,7) (2,6) Path Lenght : 4
Edges : (4,8) (5,8) (5,6) (2,6) Path Lenght : 4
********************** :
——————END OF Possible paths Information———————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 4
Number of Edges : 8
Number of nodes : 8
Number of slices from MCMCF : 5
Number of variables = 221

Objective Value : 4
——————END OF ILP Routing And Spectrum Assignment Cplex———————
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 6 - 7 - 3 : : with length 4
Slots : 1 - 2
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 7 - 8 - 4 : : with length 4
Slots : 0 - 1
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 8 - 5 - 1 : : with length 4
Slots : 3 - 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 5 - 6 - 2 : : with length 4
Slots : 2 - 3
——————BEGIN OF Forbidden Cliques Information———————
Size : 4| Omega : 4
Clique : (1 2 3 4)
———
Size : 3| Omega : 3
Clique : (1 2 3)
———
Size : 3| Omega : 3
Clique : (1 2 4)
———
Size : 3| Omega : 3
Clique : (1 3 4)
———
Size : 3| Omega : 3
Clique : (2 3 4)
———
Size : 4| Omega : 4
Clique : (1 2 3 4)
———
Size : 3| Omega : 3
Clique : (1 2 3)
———
Size : 3| Omega : 3
Clique : (1 2 4)
———
Size : 3| Omega : 3
Clique : (1 3 4)

———
Size : 3| Omega : 3
Clique : (2 3 4)
———
——————END OF Forbidden Cliques Information——————
——————BEGIN OF ILP Min Cost Multi Commodity Cplex——————
Number of demands : 4
Number of arcs : 16
Number of nodes : 8
Number of variables = 65
Capacity :2
——————END OF ILP Min Cost Multi Commodity Cplex——————
Output - MCMCF Problem -
Capacity = 2
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 7 - 8 - 4 : : with length 4
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 5 - 6 - 2 : : with length 4
——————BEGIN OF Possible paths Information——————
For demand : 1
Edges : (1,5) (5,8) (7,8) (3,7) Path Lenght : 4
Edges : (1,5) (5,6) (6,7) (3,7) Path Lenght : 4
********************** :
For demand : 2
Edges : (2,6) (5,6) (5,8) (4,8) Path Lenght : 4
Edges : (2,6) (6,7) (7,8) (4,8) Path Lenght : 4
********************** :
For demand : 3
Edges : (3,7) (6,7) (5,6) (1,5) Path Lenght : 4
Edges : (3,7) (7,8) (5,8) (1,5) Path Lenght : 4
********************** :
For demand : 4
Edges : (4,8) (7,8) (6,7) (2,6) Path Lenght : 4
Edges : (4,8) (5,8) (5,6) (2,6) Path Lenght : 4
********************** :
——————END OF Possible paths Information——————
——————BEGIN OF ILP Routing And Spectrum Assignment Cplex——————
—
Number of demands : 4

Number of Edges : 8
Number of nodes : 8
Number of slices from MCMCF : 5
Number of variables = 221
Objective Value : 4
————END OF ILP Routing And Spectrum Assignment Cplex————
Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 6 - 7 - 3 : : with length 4
Slots : 0 - 1
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 5 - 8 - 4 : : with length 4
Slots : 1 - 2
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
Slots : 2 - 3
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
Slots : 3 - 4
————BEGIN OF Forbidden Cliques Information————
Size : 4| Omega : 4
Clique : (1 2 3 4)
——
Size : 3| Omega : 3
Clique : (1 2 3)
——
Size : 3| Omega : 3
Clique : (1 2 4)
——
Size : 3| Omega : 3
Clique : (1 3 4)
——
Size : 3| Omega : 3
Clique : (2 3 4)
——
Size : 4| Omega : 4
Clique : (1 2 3 4)
——
Size : 3| Omega : 3
Clique : (1 2 3)
——
Size : 3| Omega : 3

Clique : (1 2 4)
———
Size : 3| Omega : 3
Clique : (1 3 4)
———
Size : 3| Omega : 3
Clique : (2 3 4)
———
Size : 4| Omega : 4
Clique : (1 2 3 4)
———
Size : 3| Omega : 3
Clique : (1 2 3)
———
Size : 3| Omega : 3
Clique : (1 2 4)
———
Size : 3| Omega : 3
Clique : (1 3 4)
———
Size : 3| Omega : 3
Clique : (2 3 4)
———
——————END OF Forbidden Cliques Information———————
——————BEGIN OF ILP Min Cost Multi Commodity Cplex———————
Number of demands : 4
Number of arcs : 16
Number of nodes : 8
Number of variables = 65
Capacity :4
——————END OF ILP Min Cost Multi Commodity Cplex———————
Output - MCMCF Problem -
Capacity = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 7 - 8 - 4 : : with length 4
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 8 - 5 - 1 : : with length 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
===== SOLUTION =====
This solution becomes feasible and this iteration will be stopped

Output - RSA Problem -
Interval chromatic number = 4
Routing :
From demand : 1 Between nodes : 1 to 3 : : 1 - 5 - 8 - 7 - 3 : : with length 4
Slots : 0 - 1
From demand : 2 Between nodes : 2 to 4 : : 2 - 6 - 5 - 8 - 4 : : with length 4
Slots : 1 - 2
From demand : 3 Between nodes : 3 to 1 : : 3 - 7 - 6 - 5 - 1 : : with length 4
Slots : 3 - 4
From demand : 4 Between nodes : 4 to 2 : : 4 - 8 - 7 - 6 - 2 : : with length 4
Slots : 2 - 3
—————————————————— :
Iterations : 4
—————-END OF EXECUTION——————- :