

MATHEMATICAL FORMULATIONS AND APPROACHES TO THE BRAIN TUMOUR RESECTION PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Sarp Bora İlhan
July 2024

MATHEMATICAL FORMULATIONS AND APPROACHES TO
THE BRAIN TUMOUR RESECTION PROBLEM

By Sarp Bora İlhan

July 2024

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

Taghi Khaniyev(Advisor)

Oya Karaşan

Amine Gizem Tiniç

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

MATHEMATICAL FORMULATIONS AND APPROACHES TO THE BRAIN TUMOUR RESECTION PROBLEM

Sarp Bora İlhan
M.S. in Industrial Engineering
Advisor: Taghi Khaniyev
July 2024

Brain tumors are the fifth most common tumors and the first line of treatment for a brain tumor is surgical resection. However, resecting an entire tumour mass may raise a problem when the tumour merges with healthy cells or it has formed around the brain region that is involved in vital functions. Our aim is to capture the results of resecting tumourous parts in these complex situations and selecting the tumour volume that will have the least disruption on the brain functionality. We propose three measures, namely, global efficiency, concurrent demand satisfaction and congestion, to quantify the brain functionality. We develop critical node detection based formulation for the first and multicommodity flow based models for other measures. We develop Benders decomposition and column generation algorithms to solve the mathematical models. We compare the algorithm performances with model formulations. We use a diverse range of instances to conduct detailed comparisons. Finally, we investigate the decision outputs and how they change in different cases.

Keywords: tumour resection, benders decomposition, column generation, max flow .

ÖZET

BEYİN TÜMÖRÜ ÇIKARMA PROBLEMİNE MATEMATİKSEL FORMÜLASYONLAR VE YAKLAŞIMLAR

Sarp Bora İlhan
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanı: Taghi Khaniyev
Temmuz 2024

Beyin tümörleri, en yaygın beşinci tümördür ve beyin tümörü tedavisi için ilk adım, cerrahi rezeksiyondur. Ancak tümörün sağlıklı hücrelerle birleştiği veya hayati fonksiyonları olan beyin bölgelerinin etrafında olduğu durumlarda tümör kitlesinin tamamen çıkarılması sorun yaratabilir. Amacımız, bu karmaşık durumlarda tümörlü bölgelerin çıkarılmasının sonuçlarını saptamak ve çıkarılması durumunda çevre dokular üzerinde beyin fonksiyonlitesine en az etkiye sahip olacak tümör hacmini seçmektir. Beyin fonksiyonelliğini ölçmek için global verimlilik, eşzamanlı talep karşılama ve tıkanıklık olmak üzere üç ölçü önerlmıştır. İlk ölçü için kritik düğüm tespitine dayalı bir modeller, diğer ölçüler için çoklu ürün akışı tabanlı modeller oluşturulmuştur. Matematiksel modelleri çözmek için Benders ayrışımı ve kolon oluşturma algoritmaları geliştirilmiştir. Geliştirilen algoritmalar oluşturulan matematiksel modeller ile karşılaştırmıştır. Bu deneyler için geniş bir örneklem kümesi oluşturulmuştur. Son olarak, farklı durumlarda karar çıktılarını ve farklı örneklem üzerinde nasıl değiştikleri incelenmiştir.

Anahtar sözcükler: tümör çıkarma, benders ayrışımı, kolon oluşturma, maksimum akış.

Acknowledgement

I would like to extend my deepest gratitude to my advisor Asst. Prof. Taghi Khaniyev. He has provided guidance and support throughout this research process. His expertise and dedication have greatly contributed to the completion of this thesis. I am grateful for his patience and the knowledge he has shared with me.

I would like to extend my deepest gratitude to Prof. Oya Karařan for her unwavering support, guidance, and encouragement throughout this process. Her insightful feedback have been instrumental in shaping my work. I am truly grateful for her exceptional dedication and the significant impact she has had on my research.

I would also like to express my deepest appreciation to Asst. Prof. Amine Gizem Tiniç for her invaluable jury comments. Her feedback has been crucial in refining my work.

Throughout my Master education, I received a graduate scholarship from TÜBİTAK under the BİDEB 2232 project (121C155) titled “A prescriptive approach to optimal tumor resection”. I am grateful for their invaluable support.

I could not have undertaken this journey without my dear friend, Batuhan Çelik. From the very beginning, we have laughed, cried, played games many nights, traveled, studied together, guided each other, and supported one another. Your friendship has been invaluable.

I would like to thank my mother, Alev İlhan, my father, Recep Tufan İlhan and my brother, Giray İlhan, for being there through thick and thin. I cannot imagine a world without them.

Lastly, I am sincerely grateful to Aslı Soysal for being a constant source of support and motivation. Your enduring faith in me has given me strength during difficult moments. I feel truly lucky to have you with me.

Contents

1	Introduction	1
2	Literature Review	4
2.1	Neuroscience Literature	4
2.2	Operations Research Literature	8
3	Problem Formulation and Solution Methodology	12
3.1	Global Efficiency Model	13
3.1.1	Benders Decomposition	17
3.2	Concurrent Flow Model (CFP)	23
3.2.1	Congestion Problem (CP)	33
4	Computational Results	36
4.1	Instance Generation	36
4.2	Computational Experiments	38

4.2.1	Benders Decomposition Comparison	38
4.2.2	Column Generation Comparison	45
4.3	Solution Analysis	46

5	Conclusion	53
---	------------	----



List of Figures

1.1	On the left, a healthy brain MRI [1], on the right, a brain MRI with a low grade glioma [2].	2
3.1	Example network with three nodes.	17
4.1	Example network for demand generation.	38
4.2	Average Solution time of Formulation 1 and Benders Decomposition for four instance parameters.	39
4.3	Computational comparison between CFP & Column Generation (time in seconds)	46
4.4	Global Efficiency Objective Function Values for Different β and v_T Values	48

List of Tables

3.1	Notation	14
3.2	Notations used for flow models	24
4.1	Instance parameters and their values	37
4.2	Computational comparison summary between Global Efficiency Model & Benders Decomposition (Average)	40
4.3	Computational comparison between Global Efficiency Model & Benders Decomposition for ($ N = 50$)	42
4.4	Computational comparison between Global Efficiency Model & Benders Decomposition for ($ N = 100$)	43
4.5	Computational comparison between Global Efficiency Model & Benders Decomposition for ($ N = 150$)	44
4.6	Computational comparison between CFP & Column Generation (time in seconds)	45
4.7	Solution analysis of objectives	47
4.8	Resection output in global efficiency model for ($ N = 50$)	49

4.9	Resection output in global efficiency model for ($ N = 100$)	50
4.10	Resection output in global efficiency model for ($ N = 150$)	51
4.11	Solution analysis for global efficiency results($ N = 50, \xi = 0.20$) .	52
4.12	Solution analysis for $z_{\min}$ results($ N = 50, \xi = 0.20$)	52
4.13	Solution analysis for ϕ results($ N = 50, \xi = 0.20$)	52

Chapter 1

Introduction

The human brain is the control center for both voluntary and involuntary functions. Furthermore, it is the center for important cognitive functions like problem solving, emotion regulation and consciousness. The main cells that form the parts of the brain are called neurons. Grey matter is the part that contains the neural cell bodies which regulates the mentioned vital and cognitive functions and white matter is the part that contains myelinated axons which help with the communication between different brain regions [3]. One can liken this structure to hubs and roads where the neural cell bodies are the hubs and the myelinated axons are the roads.

Like all regions of the body, the brain can succumb to cancer as well. Figure 1.1 shows a healthy brain and a brain with a tumour (abnormal cell masses). There are over 100 types of tumours classified for the brain [4]. There are more than 300,000 brain tumour cases reported each year which makes it the fifth most common cancer type. Of these total tumours 27.9% is malignant and for malignant tumours, the survival rate is 35.7% [5][6]. For such a deadly disease, the first line of treatment is resection of the tumourous regions. Resections are effective in dealing with tumours, however, a resection decision may become non-trivial if the area surrounding the core of the tumor is populated with high density of healthy cells that take part in vital cognitive functions. [7]. Operations research

tools can help aid in this decision-making problem. Representing the brain as a network where the nodes are the different brain regions (grey matter) and the edges as the connections between these regions (white matter) can help in a more informed decision making when it comes to resections.

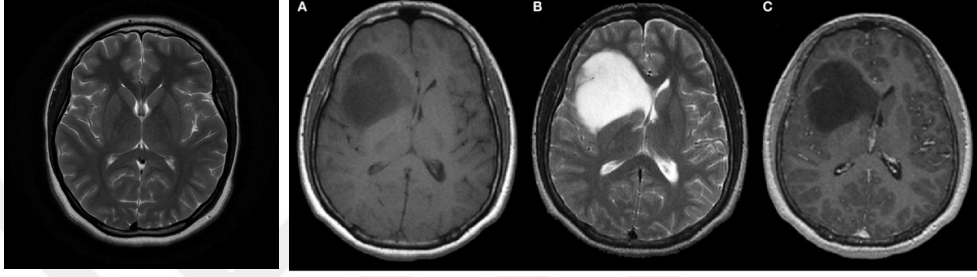


Figure 1.1: On the left, a healthy brain MRI [1], on the right, a brain MRI with a low grade glioma [2].

In this thesis, we aim to decide which tumourous part of the brain should be resected. While doing that, our objective is to resect as much volume as possible to clear the brain from the tumour as well as keep the functional loss in the brain as minimum as possible. We present several measures to quantify this functional loss. These measures have different aspects that can have importance on the patient's health.

The first measure is global efficiency which computes the average inverse shortest paths between each pair of nodes in the network. We propose a critical node detection-based model to decide which tumourous parts to be resected in order to maximize the global efficiency of the brain network after the operation (residual network). Global efficiency represents the connectivity of brain parts, and we track the impacts of resection on the brain by measuring these connections. To solve the Global Efficiency Model in large instances like brain networks, we develop a reformulation. This reformulation simplifies the model by utilizing the characteristics of the objective function. Then, we propose a Benders Decomposition Algorithm that captures the decomposable structure of the reformulation in the Benders sub-problem. To test the performance of this reformulation and the algorithm, we experiment on a variety of network instances.

The second measure we consider for quantifying the functional loss is maximum

throughput efficiency measure. We assess the brain network functionality by evaluating the quality of concurrent flow in the brain. We measure this quality by considering the brain signals from two different parts of the brain as unique commodities transferred between different parts. By taking the flow rate of each signal in the healthy brain as a pre-determined demand, we formulate a Maximum Concurrent Flow model to resect tumour that maximizes the fraction of demand satisfied for each commodity (node pair). Then, we apply a column generation algorithm to find the minimum fraction of demand satisfaction of a given network and test the performance of the column generation approach.

The third measure is the congestion measure. We quantify the damage of resection on the remaining portion of the brain with a congestion-based approach. Assuming that the brain sends the same frequency of signals before and after the resection, we don't expect the congestion in a healthy brain to be higher after the resection. Therefore, we formulate the Congestion Model to minimize congestion. We test the computation performance of these formulations and the solution algorithms, then analyze the solution outputs. Finally, we input the solution of one model to others to observe how differently these measures approach the brain tumour resection problem.

Chapter 2

Literature Review

This literature review is comprised of two sets of distinct articles. The first part contains articles relevant to the neuroscience literature and the second part contains papers relevant to the operations research literature.

2.1 Neuroscience Literature

In recent years, neuroscience research has focused, among many subjects, on enhancing brain mapping imaging methods and using machine learning to analyze imaging data for diagnostic and other purposes.

The imaging technology studies explain the connection between structural connectivity of the brain and the connections that support different cognitive functions as well as how they are changed in neurological disorders. The imaging methods can be classified under two categories: structural and functional. Magnetic Resonance Imaging (MRI) and Diffusion Tensor Imaging (DTI) are some examples to structural imaging and Functional Magnetic Resonance Imaging (fMRI) and Positron Emission Tomography (PET) are some examples to functional imaging methods [8]. Toga et al. (1998) [9] provided high resolution imaging studies with these tools (and more such as EEG, CT etc.) on this front

and established a standard for the modern imaging technologies. Conturo et al. (1999) [10] is an example to the development of advanced imaging and analysis tools as they have developed a procedure to track neural fiber pathways using Diffusion Tensory Imaging. This is an important advancement as it allows us to identify how thick the white matter tissue is and can develop our study further. The main difference between the imaging methods is that structural imaging is used to visualize the anatomical structure whereas the functional imaging techniques are used to asses brain activity and function. Both of these techniques have been crucial in the effort to map the brain.

In the pursuit of the brain mapping effort Koetter and Wanke (2005) [11] emphasized the importance of accurate and comprehensive connectivity datasets, namely, structural, functional, anatomical, electrophysiological data, and facilitating cross disciplinary collaboration and Sporns et al. (2005) [12] laid the groundwork for understanding the structural basis of brain functions and dysfunctions and emphasized the need for a comprehensive mapping of brain connectivity. These efforts were officially named under the Human Connectome Project (HCP), where the aim was to understand the structure of the brain. Stephan et al. (2001) [13] proposed a database of a macaque’s (a type of monkey) brain containing map and and connectivity data, which has been widely used and has supported the large scale brain mapping initiatives. Wedeen et al. (2005) [14] introduced the diffusion spectrum imaging. Their work presented a method for mapping the complex three dimensional structure of white matter fibers. This method improved the existing DTI by having a more precise mapping of fiber orientations and has since been widely used for white matter data.

Sporns et. al. (2007) [15] introduced an idea of identifying and classifying brain regions using a combination of structural and functional techniques on the macaque brain and a cat cerebral cortex via an index called ”small-world index” (a quantitative measure to characterize network topologies). Honey and Sporns (2008) [16] followed their effort on a study of neural connectivity by enhancing our understanding of functional integration and segregation using similar methods and showed dynamic interplay during cognitive tasks of brain regions, before

and after the placement of cortical lesions on different brain regions. These studies are similar to our studies in the sense that it is trying to determine the impact of pathologies in brain tissue. Furthermore, another study towards the same goal is done by Johansen-Berg and Behrens (2005) [17] where they have integrated probabilistic tractography (a technique used in neuroimaging to map the white matter pathways), with anatomical and functional data and improved the accuracy of brain mapping and offered new paths for exploring the brain’s structural connectivity in healthy and pathological subjects. More recently, Glasser et al. (2016) [18] made a significant contribution with a comprehensive mapping of the human cerebral cortex using the HCP data and presented a detailed parcellation of the cortex into 180 distinct areas per hemisphere. More recently Schirner et al. (2015) [19] introduced The Virtual Brain (TVB) which is a neuroinformatics tool designed to simulate large scale brain networks with individual specific data that has gained traction in recent years. Aerts et al. (2018) [20] focused on modeling brain dynamics using TVB in brain tumor patients.

Neural connectivity is a topic that is related to the mapping of the brain, where mapping focuses on identifying the structures and functions of brain regions and neural connectivity specifically examines the pathways that enable the communication between these regions as mentioned above with Honey and Sporns (2008) [16]. Bullmore and Sporns (2009) [21] further examined the connections between different brain regions. Along with the small-world index, they have added modularity and rich club indexes which are communication efficiency indexes and showed how these network characteristics support cognitive functions and their changes in neurological disorders. Furthermore, Van den Heuvel et al. (2009) [22] demonstrated that functionally linked networks are closely aligned with the structural connectivity networks which showed that regions with strong structural connections also show strong functional connectivity. In this research we try to provide a similar, albeit simpler approach to connectivity and modeling the brain as the aforementioned studies, by approaching and trying to quantify our problem through the use of networks.

Recent advancements in brain tumour modeling and resection have further shown the importance of using sophisticated methodologies. Aside from Schirner

et al. (2015) [19]’s TVB, the integration of machine learning in modeling has shown promise in enhancing the precision of modeling the brain and tumour resection. The review by Bessadok et al. (2015) [23] of the application of graph neural networks (GNN) in network neuroscience summarized the growing trend. They showed how GNN’s have been used to learn graph structures support in tasks such as disease classification. More recent studies have also created an advancement gap from the studies mentioned in Bessadok et al. (2015) [23]. Meesters et al. (2023) [24] proposed an automated pipeline using a deep convolutional neural network to segment MRI data and for the reconstruction of white matter pathways using the HCP data which significantly improved tractography (mapping the white matter pathways) accuracy and efficiency. Furthermore, Mukherjee et al. (2022) [25] developed a novel approach that combines three base generative adversarial network (GAN) models, two variants of deep convolutional GAN and Wasserstein GAN, called AGGrGAN to generate synthetic MRI scans of brain tumors. This is an important task for further growing the machine learning approaches as it provides data without class imbalance issues. The validation of the images were done on BraTS 2020 dataset, which is one of the leading brain tumour segmentation competitions that is held annually, and showed great success. Furthermore, in BraTS 2020 Isensee et al. (2020) [26] was awarded first place for the implementation of a modified U-Net called nnU-Net and achieved tumour segmentation with the predicted areas overlapping with the true values over 80% of the datasets. Liu et al. (2021) [27] explained the importance of implementing machine learning in implementation and used different deep learning models along with structural and functional data to analyze local and global effects of brain tumours (specifically glioblastomas) and connectivity changes due to tumour growth and resections. The literature on brain tumour mainly focuses on segmentation, classification and thus, diagnostics. Where our research differs is that it aims to provide a basis for a prescriptive approach for tumour removal through advanced mathematical tools.

2.2 Operations Research Literature

The relevant operations research literature for this thesis consists of critical node detection, multicommodity flow models and congestion models.

Critical node detection problem (CNDP) is important in network analysis for understanding the structures and properties of networks. The problem of CNDP is to find nodes that are important in networks, such that their removal would impact the network more than the removal of other nodes. The literature has gathered around Mixed Integer Linear Programming (MILP) formulations but recently machine learning approaches have been applied to the problem. Arulselvan et al. (2009) [28] addressed the CNDP by a MILP with an objective that will impact the remaining network's pair-wise connectivity the most. They also implemented a novel local search algorithm for finding these node groups. Granata et al. (2013) [29] introduced the concept of a critical disruption path, where their MILP objective was to minimize the largest remaining connected group of nodes through the removal of a path, which is a set of connected nodes. As a solution approach, they develop a combination of branch and price and column generation algorithms to find an optimal path. Veremyev et al. (2015) [30] focuses on a distance based CNDP where the goal is to minimize the distance based connections between nodes. The algorithm they propose is called truncate-and-resolve, where they iteratively solve MILP's to find the optimal solution for their original problem. We study a similar setting to these papers, unlike Arulselvan et al. (2009) [28] where the aim is to detect impactful nodes, like Veremyev et al. (2015) [30], we aim to find nodes that will increase efficiency and potentially minimally impact the network. Murray et al. (2007) [31] proposed a similar approach with the addition of using a flow based formulation to find critical nodes. Their model maximizes or minimizes the total flow interdicted from the network. Ventresca et al. (2018) [32] proposed a bi-objective CNDP which aims to maximize the number of connected components and minimizing the variance of the sizes of these connected groups and proposed solution methods via a multitude of evolutionary algorithms.

Differently, Yu et al. (2020) [33] proposed a machine learning approach to the CNDP in temporal networks (a network that changes its connections over time) by developing a convolutional neural network. The objective for finding the critical nodes is defined similar to Arulselvan et al. (2009) [28] and Granata et al. (2013) [29] as the most impactful nodes. Zhong et al. (2018) [34] developed a similar machine learning approach to find the critical nodes in complex networks using belief propagation, an iterative algorithm to compute marginal probabilities of nodes in networks where high probabilities refer to critical nodes. Lalou et al. (2018) [35] provided a comprehensive survey on CNDP for the articles not discussed in this review. They showcase various CNDP methodologies and compare them using various network properties. This problem has been mostly implemented to telecommunication, transportation and the defense industries. Diversly from these studies, we are analyzing the implementation of this approach on the brain networks.

Multicommodity flow problem (MCFP) is also another topic which is related to this study. Similar to the CNDP, MCFP literature has also developed around MILP formulations. Shahrokhi and Matula (1990) [36] introduced the concept of maximum concurrent flow which aims to maximize the uniform throughput across all commodity pairs in a network. This has been one of the key concepts in our formulations. Furthermore, to maximize throughput, their model utilizes an approximation scheme that uses a flow rerouting algorithm which iteratively adjusts flows based on updated distance between the nodes. Gouveia (1995) [37] introduced the hop constrained minimal spanning tree problem, where the objective is to find a minimal spanning tree such that any path from the tree's root to its branches is length constrained. They developed both undirected and directed multicommodity flow formulations for this problem. Several approaches to Lagrangean relaxation for these models were used to solve the problem faster and used computational tests to validate the effectiveness of these models. Dai et al. (2016) [38] and Dai et al. (2017) [39] proposed different path based linear programming methods for finding feasible columns in multicommodity flow problems including shortest path and widest path and implemented their methods

on different networks. Furthermore, they showed that the column generation approach to the multicommodity flow problems outperforms Lagrangian relaxation for certain instances. Moradi et al. (2015) [40] developed a bi-objective column generation algorithm for the MCFP. The main approach uses the bi-objective simplex method with a Dantzig-Wolfe decomposition. They reformulate the problem into a bi-objective master problem over capacity constraints and several single objective sub problems over the flow balance constraints. These formulations are similar to our attempt at implementing a column generation schema to the MCFP. Liu (2020) [41] introduced the concept of equilibrium pseudo flow for solving the MCFP, where the objective is to minimize the penalty associated with exceeding arc capacities. The solution approach uses a combinatorial algorithm which iteratively updates flows to find the equilibrium pseudo flow that ensures all used paths have equal minimal length. Lastly, Salimifard and Bigharaz (2020) [42] provided an extensive survey on the multicommodity flow models which includes significant papers in the literature. For further details, the interested reader is referred to the references in this paper.

The last approach to modeling our brain resection problem comes from the network congestion problem. Numerous approaches from different mathematical modeling branches are proposed to minimize network congestion. Räcke (2002) [43] offers various methods such as online routing algorithm, tree simulation and concurrent multicommodity flow problem without formulating the problem as a MILP. Furthermore, Räcke's (2008) [44] builds upon their previous work by proposing a hierarchical decomposition tree method to simplify the problem into a tree problem. This hierarchical decomposition is posed as a mixed packing and covering problem, which is a problem type that gives the upper bounds and lower bounds for the variables in the constraints, and is utilized to a convex combination of trees and eventually to find flows over this convex combination to load balance. Trees are generated with a minimum communication cost tree problem which finds tree structures that represents how traffic can flow through the network. Banner (2005) [45] considers two linear modeling problems, which are path based approaches. One of the problems restricts the amount of paths whereas the other restricts the length of the paths to distribute the flow over.

Both Räcke's (2008) [44] and Banner (2005) [45] focused on similar measures as our congestion model, however, the methods are different.



Chapter 3

Problem Formulation and Solution Methodology

The tumour resection problem is handled in this study by considering the example of low-grade glioma tumours. Low-grade gliomas grow slowly and from a central tumour mass. This results in the body forming blood vessels to the affected region and the peripheral region to be fused with healthy cells. Therefore, the removal of the entire tumor-affected region may have a larger negative impact on brain functions than if a selected subregion of the tumour is removed [46] [47]. Based on these risks, which portion of the brain should be resected poses a challenging problem. In this study, we aim to achieve maximum tumour removal while maintaining minimum damage to the function of the brain.

The problem can be approached using mathematical modeling tools. To be able to model the system, we treat the brain as a graph. In this brain graph, the connection between brain parts forms the network. Every network has two basic components which are nodes and edges. Nodes of this network represent the gray matter regions and the edges represent the white matter regions. In the brain, white matter transfers signals from different gray matter masses, thus we can visualize these signals as the flows sent through edges. First model is a critical node detection formulation improvement, and the following models are

flow models, in which they primarily differ from each other in the measure they use to quantify the brain functionality loss.

In this section, we provide three formulations for the brain tumour resection problem considering different objectives.

3.1 Global Efficiency Model

The first approach to modeling the tumour resection problem is inspired from the formulation of Veremyev et al. (2015) [30]. They propose a critical node detection formulation. The formulation uses various versions of a communication efficiency function to decide this detection problem. This is useful for the brain tumour resection problem as the global efficiency measure can be used to find the nodes that impact the communication efficiency of the brain network as least as possible while removing the tumourous regions. We want to remove as much tumourous region as possible in this formulation while minimally impacting the communication efficiency. What this means is that when a resection is completed, the remaining nodes should be able to communicate in similar hop distances from each other. Table 3.1 depicts the notation used for the global-efficiency model. Some important parameters are $f(l)$ and v_i . $f(l)$ is a pairwise connectivity function used in calculating the global efficiency. In our case we use $f(l) = \frac{1}{l}$ where l is the shortest path distance between the pair of nodes. v_i is the expected tumour volume in the i^{th} region of the brain.

Table 3.1: Notation

Symbol	Definition
Sets	
N	set of nodes
H	set of hops of paths, $H = \{0, 1, 2, \dots, L\}$
E	set of existing edges in the graph
T	set of tumourous nodes ($T \subset N$)
Parameters	
$f(l)$	connectivity function with respect to hop length $l \in H$, $f(l) = \frac{1}{l}$
v_i	volume of node $i \in N$
β	percentage of the retained brain volume $i \in N$
v_0	Upper bound on the total volume of the brain after resection ($v_0 = \sum_{i \in N} v_i \beta$)
a_{ij}	$= \begin{cases} 1, & \text{if nodes } i \in N \text{ and } j \in N \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}$
Decision variables	
x_i	$= \begin{cases} 1, & \text{if node } i \in N \text{ is resected} \\ 0, & \text{otherwise} \end{cases}$
u_{ij}^l	$= \begin{cases} 1, & \text{if there is a path of length at most } l \in H \text{ connecting} \\ & \text{nodes } i \in N \text{ and } j \in N \text{ after resection process} \\ 0, & \text{otherwise} \end{cases}$

An approach implemented for the same purpose can be seen in Khaniyev et al. (2024) [48] where their model’s aim is to maximize volume of resection while keeping the global efficiency above a predetermined level. This formulation values the brain’s functionality first and tries to resect as large section as possible. However, global efficiency is one possible metric to measure brain connectivity, and finding a desirable lower bound is not a straightforward decision. On the other hand, tumourous nodes and their threat to the brain are relatively easier to quantify. Therefore, depending on the position of the tumour and other patient-specific characteristics of the tumour may require a certain amount to be resected. Based on these reasons, we continue with our formulation, where the global efficiency is maximized and the residual tumour volume is restricted to be below a pre-defined threshold.

The aforementioned formulation provides a functional way of describing the network and connection between nodes in terms of hops, which is aligned with our assumption of interest range in communication. However, when the formulation is projected to a brain topology, which is a large and complex network, it is impossible for commercial solvers to find a solution in larger instances. Hence, solving as well as creating the model becomes computationally challenging. Thus, we need a structure that is decomposable into many other small problems. Accordingly, we present a more efficient formulation for the same problem presented in Khaniyev et al. (2024) [48].

Global Efficiency Model

$$\max \sum_{i \in N} \sum_{j \in N} \sum_{l=1}^L f(l)(u_{ij}^l - u_{ij}^{(l-1)}) \quad (3.1)$$

s.t.

$$v_0 \geq \sum_{i \in N} v_i(1 - x_i) \quad (3.2)$$

$$u_{ij}^l \leq \sum_{k=1}^{l-1} u_{ij}^k + \sum_{t \in N: t \neq i, j} a_{it} u_{tj}^{(l-1)} \quad \forall i, j \in N, i \neq j, \forall l \in \{2, \dots, L\} \quad (3.3)$$

$$u_{ij}^1 \leq a_{ij} \quad \forall i, j \in N, i \neq j \quad (3.4)$$

$$u_{ij}^l \leq 1 - x_i \quad \forall i, j \in N, i \neq j, \forall l \in \{1, \dots, L\} \quad (3.5)$$

$$u_{ij}^l \leq 1 - x_j \quad \forall i, j \in N, i \neq j, \forall l \in \{1, \dots, L\} \quad (3.6)$$

$$x_i \in \{0, 1\} \quad \forall i \in N \quad (3.7)$$

$$x_i = 0 \quad \forall i \in N \setminus T \quad (3.8)$$

$$1 \geq u_{ij}^l \geq 0 \quad \forall i, j \in N, \forall l \in \{1, \dots, L\} \quad (3.9)$$

In this model, the goal is to maximize global efficiency (3.1) while restricting the remaining volume below a threshold (v_0) with constraint (3.2). v_0 generally represents a patient specific percentage of initial network volume. In the tumour resection setting, considering that each tumourous node and its volume are known, one can find efficient solutions with solving the model for different levels of v_0 .

Constraint (3.3) forces u_{ij}^l to be 0 if there is no shorter path from i to j or a path with exactly of length l passing through an adjacent node of i (t). Otherwise, u_{ij}^l will tend to be 1 due to the objective function direction. In the model, if there is an edge between two nodes, we can accept that longer paths also exist. The model ensures this relation again by constraint (3.3).

For example, in Figure 3.1, node 2 is adjacent to both nodes 1 and 3, but nodes 1 and 3 are not adjacent to each other. In this graph one can find a path

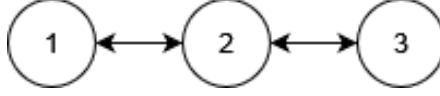


Figure 3.1: Example network with three nodes.

of length 2 from 1 to 3 by passing through node 2. The first part of the RHS of constraint (3.3) counts the number of existing paths from node i to node j with length up until $l - 1$ whereas the second part of the RHS counts the exists a path from adjacent of i (t) to j . Then, both of these sections form an upper bound to u_{ij}^l .

Furthermore, constraint (3.4) sets upper bounds to path decisions with length 1 (direct adjacency). u_{ij}^l cannot exceed the $\{i, j\}^{th}$ element of adjacency matrix (by (3.4)). Following this, any path between nodes i and j is restricted to 0 by (3.5) and (3.6) if either of these nodes are resected from the brain.

Finally, constraint (3.7) and (3.8) are binary restrictions and (3.9) is the domain constraint for the u_{ij}^l . Further notice that u_{ij}^l variables can be taken as continuous variables, and due to the objective function, the optimal will be an integer [48].

In order to solve this reformulated model for large instances, we have implemented a Benders decomposition algorithm which will be discussed next.

3.1.1 Benders Decomposition

Benders decomposition is an algorithm mainly used to solve large-scale linear and mixed integer programs faster. It separates the problem into a more manageable master problem and sub problem(s). The master problem includes the complicating variables, usually the integer or binary variables and the constraints associated with only these variables. Then the rest of the constraints are handled at sub problem. Master problem is solved to find values for integer variables which are used in the sub problem as parameters. Afterwards, sub problem finds the best solution for given integer decisions. These solutions generate a Benders

cut which is added to the master problem.

The master problem considers only a subset of the constraints and the remaining constraints are converted into cutting planes for given integer variables. Moreover, if the sub problem finds a feasible solution, the found continuous solutions and the integer solution for the master problem is feasible to original problem. Therefore, for a maximization (minimization) problem, the master problem provides an upper (lower) bound and the sub problem provides a lower (upper) bound. The master and sub problems are solved iteratively until the lower and upper bounds converge with predefined precision ϵ . Benders decomposition is particularly useful for problems that have a block diagonal structure in the sub problem which makes parallel processing possible.

The Global Efficiency Model includes both integer and continuous variables. The objective, global efficiency, is the function of continuous variables only. There is a single integer-only constraint, which is the restriction about resection volume. This means that the master problem starts as a very simple problem. Additionally, there are some sub problem properties that enable Benders decomposition to be an effective tool for this problem. For an arbitrary feasible solution obtained in the master problem, there exists a feasible solution in the sub problem. Secondly, the sub problem can be decomposed into $|N|$ sub problems due to the block diagonal structure.

As we apply a Benders decomposition, the Master Problem is as follows:

Master Problem

$$\max \sum_{j \in N} \theta_j \quad (3.10)$$

s.t.

$$v_0 \geq \sum_{i \in N} v_i(1 - x_i) \quad (3.11)$$

$$\theta_j \leq \sum_{i \in N, i \neq j} a_{ij} \lambda_i^\tau + \sum_{i \in N, i \neq j} \sum_{l \in H} (1 - x_i) \mu_i^{l\tau} + \sum_{i \in N, i \neq j} \sum_{l \in H} (1 - x_j) \beta_i^{l\tau} \quad \forall j \in N \quad (3.12)$$

$$x_i \in \{0, 1\} \quad \forall i \in T \quad (3.13)$$

$$x_i = 0 \quad \forall i \in N \setminus T \quad (3.14)$$

and for an arbitrary $j \in N$, the Sub Problem j under given values of x_i s is:

Sub Problem j

$$\max \sum_{i \in N} \sum_{j \in N} \sum_{l=1}^L f(l)(u_{ij}^l - u_{ij}^{(l-1)}) \quad (3.15)$$

s.t.

$$u_{ij}^1 \leq a_{ij} \quad \forall i \in N \quad (3.16)$$

$$u_{ij}^l \leq 1 - x_i \quad \forall i \in N, \forall l \in \{1, \dots, L\} \quad (3.17)$$

$$u_{ij}^l \leq 1 - x_j \quad \forall i \in N, \forall l \in \{1, \dots, L\} \quad (3.18)$$

$$u_{ij}^l \leq \sum_{k=1}^{l-1} u_{ij}^k + \sum_{t \in N: t \neq i, t \neq j} a_{it} u_{tj}^{(l-1)} \quad \forall i \in N, \forall l \in \{2, \dots, L\} \quad (3.19)$$

$$u_{ij}^l \geq 0 \quad \forall i \in N, \forall l \in \{1, \dots, L\} \quad (3.20)$$

Decomposing the network over j means that each divided sub problem optimizes the problem related to the destination node. Furthermore, analyzing the dual of the decomposed sub problem is important for the purposes of implementation. Let $\lambda_i, \mu_i^l, \beta_i^l$ and γ_i^l represent the dual variables for the constraints (3.16), (3.17), (3.18) and (3.19), respectively. The constraint sets (3.22), (3.23) and (3.24) are dual constraints for variables u_{ij}^l for $l = 1, l > 1$ and $l = L$ respectively. Notice (3.22) and (3.23) both have γ_i^l and $\gamma_i^{(l+1)}$. For clarification, this comes from the constraint set (3.19) as both u_{ij}^l and $u_{ij}^{(l-1)}$ variables are found in the same rows, thus u_{ij}^l and $u_{ij}^{(l+1)}$ variables in the same columns.

Dual Sub Problem j

$$\min \sum_{\substack{i \in N, \\ i \neq j}} a_{ij} \lambda_i + \sum_{\substack{i \in N \\ i \neq j}} \sum_{l \in H} (1-x_i) \mu_i^l + \sum_{\substack{i \in N \\ i \neq j}} \sum_{l \in H} (1-x_j) \beta_i^l \quad (3.21)$$

s.t.

$$\lambda_i + \mu_i^1 + \beta_i^1 - \sum_{h=2}^L \gamma_i^h - \sum_{t \in N, t \neq i, t \neq j} a_{it} \gamma_t^2 \geq f(1) - f(2) \quad \forall i \in N \quad (3.22)$$

$$\mu_i^l + \beta_i^l + \gamma_i^l - \sum_{h=l+1}^L \gamma_i^h - \sum_{t \in N, t \neq i, t \neq j} a_{it} \gamma_t^{l+1} \geq f(l) - f(l+1) \quad \forall i \in N, \forall l \in \{2, \dots, L-1\} \quad (3.23)$$

$$\mu_i^L + \beta_i^L + \gamma_i^L \geq f(L) \quad \forall i \in N \quad (3.24)$$

$$\lambda_i \geq 0 \quad \forall i \in N \quad (3.25)$$

$$\mu_i^l \geq 0 \quad \forall i \in N \forall l \in \{1, \dots, L\} \quad (3.26)$$

$$\gamma_i^l \geq 0 \quad \forall i \in N \forall l \in \{2, \dots, L\} \quad (3.27)$$

Finally, let τ be the iteration counter and $\lambda_i^\tau, \mu_i^{l\tau}$, and $\beta_i^{l\tau}$ be the values of dual variables at iteration τ ; Benders cut added to Master Problem as can be seen above, generated by the solution of Sub Problem j at iteration τ is:

$$\theta_j \leq \sum_{i \in N, i \neq j} a_{ij} \lambda_i^\tau + \sum_{i \in N, i \neq j} \sum_{l \in H} (1-x_i) \mu_i^{l\tau} + \sum_{i \in N, i \neq j} \sum_{l \in H} (1-x_j) \beta_i^{l\tau} \quad \forall j \in N$$

The implementation of this decomposition is rather straightforward as a result of the structure of the model. Firstly, the problem of finding a feasible initial solution is easily solved by making the x_i variables equal to 1 as it will provide the worst solution possible. Secondly, the master problem requires the cuts from the iteration τ to be added separately for each sub problem. Finally, rather than constructing the sub problems over and over again as we would need to alter the variables x_i , we can only change the dual problems objective functions. All of these attributes allow for the ease of implementation. Furthermore, Algorithm 1 shows the pseudocode of the implemented algorithm.

Algorithm 1 Benders Decomposition Algorithm

```
1: Step 1: Initialization
2:  $x_i^{(0)} = 1$  for all  $i$  in  $C$ 
3:  $\tau = 0$ ,  $LB = 0$ , and  $UB = 1$ 
4: Step 2: Sub Problems
5:  $\tau = \tau + 1$ 
6: for  $j \in N$ 
7:   if  $x_j^\tau \neq 1$ 
8:     Solve Dual Sub Problem  $j$  with  $x_i^\tau$  values
9:     Generate cutting plane:
10:    
$$\theta_j \leq \sum_{i \in N, i \neq j} a_{ij} \lambda_i^\tau + \sum_{i \in N, i \neq j} \sum_{l \in H} (1 - x_i) \mu_i^{l\tau} + \sum_{i \in N, i \neq j} \sum_{l \in H} (1 - x_j) \beta_i^{l\tau}$$

11:   end if
12: end for
13: Calculate  $\sum_i$  dual sub-problem objective value ( $\rho$ )
14: if  $\rho \geq LB$ 
15:    $LB = \rho$ 
16: end if
17: Step 3: Master Problem
18: Add cutting planes to Master Problem
19: Solve Master Problem
20: if  $\sum_{i \in N} \theta_i \leq UB$ 
21:    $UB = \sum_{i \in N} \theta_i$ 
22: end if
23: Step 4: Termination
24: if  $|UB - LB| \leq \epsilon$ 
25:   return Global Efficiency ( $= \sum_{i \in N} \theta_i$ ) and  $x_i^\tau$  for  $i \in N$ 
26: end if
27: else
28:   Go Step 2
29: end else
```

3.2 Concurrent Flow Model (CFP)

Global efficiency-based models are effective in modeling the brain, yet these models only account for the shortest path lengths between nodes. However, the white matter thicknesses/density connecting the nodes are a vital part of the connections in the brain, representing the capacity of the channel for information flow. Alternative to the global efficiency model, we propose flow-based modeling approaches that incorporates this capacity information to model the brain tumour resection problem.

Any brain function is a result of electrical signals between different regions (nodes) of the brain. Although multiple signals from/to different parts of the brain may be required to generate a function, we assume that each signal is independently important. Therefore, when modeling the brain network with a flow-based modeling approach, we should consider multicommodity flow where, each signal would represent a unique unit of flow of a specific commodity, and the communicating pairs represent source and destination nodes. Furthermore, since we would like to include the white matter thicknesses/density in the formulation, we can achieve this through the addition of capacities to the edges.

The second measure calculates the maximum possible minimum percentage of demand satisfied among all pairs in the network. This is aimed to quantify the information distribution capability of the network. The capacity interpretations are as explained above, and the demand can be interpreted and generated in various ways. Our demand instance prioritizes the adjacency and capacities between the node pairs. The demand generation instances are explained in detail in Chapter 4. Using the same example in Figure 3.1, let the capacities of the edges to be 1 and 2, respectively from left to right. Each of the node pairs have demands to and from each other, where the values can be different. In this case, let us assume there are only the demand pairs $\{1,2\}$, $\{2,1\}$, $\{1,3\}$ and $\{3,1\}$ with values 0.5, 0.5, 0.25 and 0.5 respectively. Since the edge between node 1 and 2 has capacity value of 1 and needs to accommodate for all of the aforementioned demand, the edge is overburdened. Since not all of the demand can be satisfied, the measure keeps

Table 3.2: Notations used for flow models

Symbol	Definition
Sets	
N	set of nodes
E	set of existing edges in the graph
T	set of tumourous nodes
P	set of paths for the column generation algorithm (paths meaning a connected set of nodes without cycles)
Parameters	
v_i	volume of node $i \in N$
v_0	Upper bound on the total volume of the brain after resection ($v_0 = \sum_{i \in N} v_i \psi$ for some $1 \geq \psi \geq 0$)
C_{ij}	capacity of edge $\{i, j\} \in E$
C_e	capacity of edge $e \in E$ ($C_e = C_{\{i,j\}} = C_{ij} = C_{ji}$)
d_{ij}	demand between of node $i \in N$ and $j \in N$
a_{ij}	$= \begin{cases} 1, & \text{if nodes } i \in N \text{ and } j \in N \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}$
a_e^p	$= \begin{cases} 1, & \text{if edge } e \in E \text{ is on path } p \in P \\ 0, & \text{otherwise} \end{cases}$
b_{ij}^p	$= \begin{cases} 1, & \text{if path } p \in P \text{ is from node } i \in N \text{ to node } j \in N \\ 0, & \text{otherwise} \end{cases}$
M	a large number
Decision variables	
x_i	$= \begin{cases} 1, & \text{if node } i \in N \text{ is resected} \\ 0, & \text{otherwise} \end{cases}$
y_{ij}^{kl}	flow amount from $i \in N$ to $j \in N$ with commodity source $k \in N$ at hop $l \in H$
y^p	flow amount that passes through path $p \in P$
$z_{\min}$	ratio of the demand satisfied / throughput
ϕ	maximum congestion in the network

track of the percentages of the demands that are satisfied. The optimal fraction in this case is $\frac{1}{0.5+0.5+0.25+0.5} = 0.5714$. This value is the maximum possible minimum percentage of demand satisfied for this network because any decrease in any of the flows to achieve a better percentage of demand satisfaction for other flows would make the decreased flow the new minimum percentage of demand satisfied. As an example, if the flow from 1 to 2 was to be 0 instead of 0.5, then the other flows would be higher but the minimum percentage of demand satisfied is recorded as 0, since there is no flow where there should be for one pair.

We decide the optimal resection considering the second measure by solving a concurrent flow problem. The Concurrent Flow Problem is a special case of the Multicommodity Flow Problem, with flows represented by the fraction of the demand. The formulation is inspired by the paper by Shahrokhi and Matula (1990) [36]. We added the hop constraints to fulfill the that two brain regions are practically non-communicating, if the shortest path between them has a length more than l hops. The formulation of this measure using the CFP is as follows:

Concurrent Flow Model

$$\max \quad z_{\min} \quad (3.28)$$

s.t.

$$\sum_{j \in N} a_{ij} y_{ij}^{i1} = \sum_{j \in N} z_{ij} d_{ij} \quad \forall i \in N \quad (3.29)$$

$$\sum_{l=2}^{L+1} \sum_{j \in N} a_{ij} (y_{ji}^{k(l-1)} - y_{ij}^{kl}) = z_{ki} d_{ki} \quad \forall i \in N, \forall k \in N, k \neq i \quad (3.30)$$

$$\sum_{l \in L} \sum_{k \in N} a_{ij} (y_{ij}^{kl} + y_{ji}^{kl}) \leq C_{ij} \quad \forall i \in N, \forall j \in N \quad (3.31)$$

$$\sum_{j \in N} (z_{ij} + \sum_{k \in N} \sum_{l \in L} y_{ij}^{kl}) \leq M(1 - x_i) \quad \forall i \in N \quad (3.32)$$

$$z_{\min} \leq z_{ij} + x_i + x_j \quad \forall i \in N, \forall j \in N \quad (3.33)$$

(3.2), (3.7)

$$\sum_{j \in N} a_{ij} (y_{ji}^{k(l-1)} - y_{ij}^{kl}) \geq 0 \quad \forall i \in N, \forall k \in N, \forall l \in \{2, \dots, L+1\} \quad (3.34)$$

$$y_{ij}^{k(L+1)} = 0 \quad \forall i \in N, \forall j \in N, \forall k \in N \quad (3.35)$$

$$y_{ij}^{kl} \geq 0 \quad \forall i \in N, \forall j \in N, \forall k \in N, \forall l \in L \quad (3.36)$$

$$1 \geq z_{\min} \geq 0 \quad (3.37)$$

$$1 \geq z_{ij} \geq 0 \quad \forall i, j \in N \quad (3.38)$$

The objective is to maximize the minimum fraction of satisfied/covered demand of all demands, which is represented by the decision variable $z_{\min}$. Constraint (3.29) and (3.30) satisfies the flow balance of the network where (3.29) ensures that the source node sends as much flow to the adjacent nodes equal to the total amount of demand that can be satisfied of the source nodes' commodities and (3.30) ensures that the incoming flow from the source node k via the node j to node i satisfies z_{ij} times the demand required from source node k at node i . In both cases, a_{ij} ensures that the flow balance is only valid for existing edges.

(3.31) is the capacity constraint of each edge. As the flows can come from both directions on an edge, the sum is over both $\{i,j\}$ and $\{j,i\}$ and should not be above the limit C_{ij} . Constraints (3.32) and (3.33) are the linking constraints of resection and the flow. (3.32) forbids any flow passing through the resected node and sets the demand satisfaction of the resected node to 0. (3.33) counts only the demand satisfaction of unresected nodes while computing the concurrent satisfaction rate.

The path connectivity from source to destination is constructed by constraint (3.34). Any flow coming to node j can leave the node only after incrementing the hop length. This increment is considered in constraint (3.30); however, (3.34) ensures the limit. Moreover, (3.29) initializes the hop length from the source as 1. Constraint (3.35) forces the auxiliary variables to be 0. (path with length longer than L hop is not in our interest.) (3.36) - (3.38) are the nonnegativity constraints.

Furthermore, note that the flow information (y_{ij}^{kl}) is described based on the edge on which the flow is happening, the source of the commodity's flow, and the current hop length in the path. On the other hand, because each demand is defined for a predetermined (source, destination) pair, one can consider this pair as a unique commodity. Consequently, y_{ij}^{kl} denotes the aggregate flow of multiple commodities from the same source node k , despite the destination is not explicitly shown.

Each feasible solution (optimal or not) to a flow problem can be represented by paths from source nodes to destination nodes. This property is guaranteed by flow balance constraints. Based on this, we can develop a path-based formulation for CFP. However, defining all paths as decision variables is not practical. To answer that, we use a column-generation algorithm to create necessary paths and improve the solution gradually.

Before going into details about our specific application, the column generation algorithm is a computation method to solve large-scale linear programming problems efficiently. This large scale is caused by the number of solutions. The

resection decision is a binary decision; therefore, we solve the linear relaxation of the problem and apply branch and bound algorithm if it is necessary. Although we solve the linear relaxation, the problem can be challenging due to dimensionality. To address this computational complexity, the column generation algorithm starts with a small subset of decision variables called columns. Then, it finds the optimal solution for the reduced model. This model is called the Master problem. The optimal solution of the master problem does not guarantee the optimality in the original problem.

To prove optimality or to find a variable (column) that can improve the solution, a new model is solved. This problem is called the Sub Problem or the Pricing problem. The pricing problem constraints ensure that the generated column represents a valid variable to the original problem. Moreover, the objective function of the sub-problem is the equation of the complementary slackness. Therefore, the objective function value gives the result of whether adding the new column can improve the solution or not. Finally, the algorithm solves the master problem and pricing problem iteratively until the pricing problem cannot find a better column.

In CFP setting, columns are possible paths from a source to a destination. These columns are represented by the index set P . P increases with each generated column. Then, column information is stored by a_e^p and b_{ij}^p parameters. a_e^p takes the value 1 if path p includes edge $e \in E$; otherwise, it is 0. b_{ij}^p takes the value 1 if path p serves the demand of commodity from i to j ; otherwise it is 0. Furthermore, these parameters will be addressed later in the pricing problem. Decision variable y^p is the amount of flow passing through path p . These paths are used in the following Master Problem:

Column Generation (Restricted) Master Problem

$$\max \quad z_{\min} \tag{3.39}$$

s.t.

$$\sum_{p \in P} a_e^p y^p \leq C_e(x_i + x_j) \quad \forall e \in E, \{i, j\} = e \tag{3.40}$$

$$\sum_{p \in P} b_{ij}^p y^p = z_{\min} d_{ij} \quad \forall i \in N, \forall j \in N \tag{3.41}$$

$$v_0 \geq \sum_{i \in N} v_i(1 - x_i) \tag{3.42}$$

$$1 \geq x_i \geq 0 \quad \forall i \in N \tag{3.43}$$

$$y^p \geq 0 \quad \forall p \in P \tag{3.44}$$

$$1 \geq z_{\min} \geq 0 \tag{3.45}$$

The objective function (3.39) is still described by variable $z_{\min}$. Since $z_{\min}$ is a function of columns, it is not explicitly present in the columns. Constraint (3.40) is equivalent to constraint (3.31) as it sets the capacity for total flow sent on edge i . But now, it forbids any flow through a path if there exists a resected node on the path. The demand satisfaction rate is calculated by constraint (3.41). Finally, the remaining constraints are nonnegativity of the variables.

Let, σ_e , η_{ij} , and π be the dual variables corresponding to constraints (3.40) and (3.41), respectively. The dual of the restricted master problem is as flows:

Column Generation Master Dual Problem

$$\min \sum_{e \in E} C_e \sigma_e \quad (3.46)$$

s.t.

$$\sum_{e \in E} a_e^p \sigma_e + \sum_{i \in N} \sum_{j \in N} b_{ij}^p \eta_{ij} \geq 0 \quad \forall p \in P \quad (3.47)$$

$$\sum_{i \in N} \sum_{j \in N} -d_{ij} \eta_{ij} \geq 1 \quad (3.48)$$

$$\sum_{e \in E} \sigma_e + \pi \geq 0 \quad (3.49)$$

$$\sigma^p \geq 0 \quad \forall p \in P \quad (3.50)$$

$$\eta_{ij} : urs \quad \forall i \in N, \forall j \in N \quad (3.51)$$

When we solve the master problem for a subset of paths, we ignore the constraints in the restricted dual problem corresponding to paths that are not in the restricted master problem. Therefore if the optimal solution to the restricted dual problem is feasible for the dual of the original Master Problem, the optimal solution we found in the restricted master problem is optimal for the original problem.

To check the optimality, we search for a violating constraint. There may be multiple constraints violated; however, in the pricing problem, we choose the one that is violated most by minimizing the objective function (3.52). If the objective value is less than 0, then there exists at least one violated constraint. Constraints (3.53), (3.54), and (3.55) are flow balance constraints for source, sink, and transition nodes, respectively. Furthermore, the path corresponding to this violated constraint must have at most L hop length; hence we add constraint (3.56). The pricing problem is the following:

Column Generation Pricing (Sub) Problem

$$\min \sum_{i \in N} \sum_{j \in N} a_{ij} \sigma_{ij} + \sum_{i \in N} \sum_{j \in N} b_{ij} \eta_{ij} \quad (3.52)$$

s.t.

$$\sum_{j \in N} a_{ij} = \sum_{j \in N} b_{ij} \quad \forall i \in N \quad (3.53)$$

$$\sum_{i \in N} a_{ij} = \sum_{i \in N} b_{ij} \quad \forall j \in N \quad (3.54)$$

$$\sum_{j \in N} a_{ij} - \sum_{j \in N} a_{ji} = 0 \quad \forall i \in N \quad (3.55)$$

$$\sum_{i \in N} \sum_{j \in N} a_{ij} \leq L \quad (3.56)$$

$$a_{ij} \geq 0 \quad \forall i \in N, \forall j \in N \quad (3.57)$$

$$b_{ij} \in \{0, 1\} \quad \forall i \in N, \forall j \in N \quad (3.58)$$

Note that, we can solve this pricing problem for each commodity individually. By doing this, instead of generating one column at a time, we can find multiple paths in each iteration. Additionally, for a specific commodity $\{i, j\}$, b variable takes values 1 and 0 for $\{i, j\}$ pair and other pairs, respectively. Hence they become parameters. As a result, the pricing problem turns into a shortest-path problem with additional hop limitation constraint (3.56). a_{ij} for path p becomes the parameter a_e^p where $i, j = e$.

In the implementation of this method, we generate all shortest paths up to 3 hops and use these paths as the initial set of columns in the restricted master problem. Furthermore, the pricing problem is even easier to solve since there are existing shortest path algorithms, optimized for speed, and we do not need to solve a mathematical model. Instead, we employ an “all shortest paths algorithm” on a network where the unique edges are the same, however, instead of the capacity values, the capacities of the edges are the values of the variable σ_e . This way the all shortest path algorithm, solved on this graph and after the η_{ij} values are

added to each found shortest path, can check whether (3.52) is less than 0 or not. The pseudocode of our column generation implementation is presented in Algorithm 2. Let τ be the iteration count and P^τ be the set of paths at iteration τ .

Algorithm 2 Column Generation Algorithm

```

1: Step 1: Initialization ( $\tau = 0$ )
2: Execute “all shortest paths algorithm” for 3 hops
3: Add shortest paths to  $P^\tau$ 
4: Step 2: Restricted Master Problem
5:  $\tau = \tau + 1$ 
6:  $P^{(\tau+1)} = P^\tau$ 
7: Solve the restricted master problem with columns  $P^\tau$ 
8: Find dual Variables
9: Step 3: Column Generation
10: Execute “all shortest paths algorithm” for  $\leq |L|$  hops
11: for Node  $i \in N$ 
12:   for Node  $j \in N, (j \neq i)$ 
13:     Assign  $a_{kl}$  values for shortest path from  $i$  to  $j$ 
14:     if  $\sum_{k \in N} \sum_{l \in N} (a_{kl} \sigma_{kl}^\tau) + \eta_{ij}^\tau \leq 0$ 
15:       Add the shortest path to path set  $P^{(\tau+1)}$ 
16:     end if
17:   end for
18: end for
19: Step 4: Termination
20: if  $P^{(\tau+1)} = P^\tau$ 
21:   return  $z_{\min}$  and  $y^{p^\tau}$ 
22: end if
23: else
24:   Go Step 2
25: end else

```

In column generation, we solve the linear relaxation of the problem; hence, we have to check the integrality of x variables and apply a branch and bound algorithm if necessary. However, The column generation algorithm fails to decide nodes to be resected in a reasonable time even with linear relaxation, which will be explained in Chapter 4. Therefore, we do not continue with the branch and bound algorithm. Instead, we also test the column generation for a simplified problem by taking CFP formulation and removing constraints related to resection. With these changes, the formulation does not decide which tumourous nodes should

be resected but instead measures a given network. Elimination of the resection decision makes the problem linear and the path-edge approach easier.

3.2.1 Congestion Problem (CP)

Minimizing congestion has mostly been studied in the context of traffic congestion or signal congestion in telecommunication networks, which requires the model to span for multiple intervals to track congestion. However, since this approach aligns with the concept of optimizing neural pathways for signal transmission we chose to adopt it as well. This aim of this measure is to quantify the communication speed between the parts of the brain, as congestion of signals in the brain grows, the speed of communication decreases [49].

Again, using the same example above in Figure 3.1 with the same demand and capacity values defined in the CFP example, the calculation of the congestion measure can be demonstrated. As all of the flow is being sent through the edges, the congestion is calculated for the edges to be $\frac{0.5+0.5+0.25+0.5}{1} = 1.75$ and $\frac{0.25+0.5}{2} = 0.375$, respectively from right to left. The edge that connects 1 and 2 is overused, creating the most congestion in the network whereas the flows on the edge between 2 and 3 do not use the capacity to its fullest. In this instance the congestion is reported as 1.75, since the edge between 1 and 2 generate the most congestion.

Unlike Global Efficiency Problem and CFP, CP maintains the existing communication between nodes and minimizes the workload on white matters after resection. Hence, the model satisfies all the demand even if the shortest path from source to destination exceeds hop length L after resection. Still, the interest range assumption is considered while deciding demand parameters for the original network. Consequently, we substitute y_{ij}^{kl} with y_{ij}^k as hop information is not necessary. The formulation is as follows:

Congestion Model

$$\min \quad \phi \tag{3.59}$$

s.t.

$$\sum_{j \in N} a_{ij} y_{ij}^i = \left(\sum_{j \in N} d_{ij} \right) (1 - x_i) \quad \forall i \in N, \tag{3.60}$$

$$\sum_{j \in N} a_{ij} (y_{ji}^k - y_{ij}^k) = d_{ki} (1 - x_i) \quad \forall i \in N, \forall k \in N, k \neq i \tag{3.61}$$

$$\frac{1}{C_{ij}} \sum_{k \in N} a_{ij} y_{ij}^k \leq \phi \quad \forall i \in N, \forall j \in N \tag{3.62}$$

$$y_{ij}^k + y_{ji}^k \leq M(1 - x_i) \quad \forall i \in N, \forall j \in N, \forall k \in N \tag{3.63}$$

$$v_0 \geq \sum_{i \in N} v_i (1 - x_i) \tag{3.64}$$

$$x_i \in \{0, 1\} \quad \forall i \in N \tag{3.65}$$

$$y_{ij}^k \geq 0 \quad \forall i \in N, \forall j \in N, \forall k \in N \tag{3.66}$$

$$\phi \geq 0 \tag{3.67}$$

The objective function, ϕ is the congestion of the network. This congestion is calculated over the edge with the highest $\frac{\text{Required capacity}}{\text{Existing capacity}}$ rate. Constraints (3.60), (3.61) and (3.62) works similar to (3.29), (3.30) and (3.31).

(3.60) and (3.61) are flow constraints similar to the ones explained in (3.29) and (3.30). The only difference is the exclusion of the variable $z_{\min}$ and inclusion of y_i . (3.60) is the flow balance for the source node and (3.61) is the flow balance for transition and destination nodes over the nodes that are decided not to be resected.

(3.62) calculates the congestion of the network based on the total flow passed through each edge and their respective capacities.

Constraint (3.63) is the flow-resection linking constraint that forbids any flow on an edge that connects a resected node with another node since eliminating the

node also eliminates all the edges from/to that node. Finally, (3.66) and (3.67) are nonnegativity constraints for flow and congestion decisions.



Chapter 4

Computational Results

4.1 Instance Generation

We created a random instance set with varying parameters and different values set to them shown in Table 4.1. $|N|$ is the number of nodes in the graph. $1 - \xi$ is the sparsity of the graph; high sparsity means there is less number of edges connecting the nodes. v_T represents the fraction of tumour volume to brain volume. β is the percentage of volume remaining in the brain after the resection. By taking the combinations of these settings, we have 108 instances to investigate diverse cases in our experiments. Generating certain parameters for a brain network is inherently complex, and the algorithm must return specialized results for each patient. Instead, we generate capacity and demand parameters for our instances based on the observed positive correlation between the interaction density of two gray matter regions and the thickness of the white matter connecting them [50]. For example, if the white matter between arbitrary nodes i, j is two times thicker than the one between arbitrary nodes m, n then the capacity (C) relation between these edges is: $C_{ij} = 2C_{mn}$.

For our instances, we assign capacities to edges randomly ranging from 1 to 10. Then, starting with an arbitrary node in the network, we initialize the supply

Table 4.1: Instance parameters and their values

Instance Parameter	Value Settings			
# of Nodes ($ N $)	50	100	150	
Sparcity Level ($1 - \xi$)	0.8	0.75	0.7	
Tumour Size (v_T)	0.15	0.2	0.25	
β	0.85	0.9	0.95	1

available to send by adding the capacities of connecting edges. These supplies are signals sent to nodes around. We transfer this supply to each adjacent node proportional to the capacities of the edges. For example, in Figure 4.2 of a partial network, the node i has 3 adjacent nodes, which are j , k , and l . The capacities of the edges connecting i are 5, 10, and 10, respectively. Thus, node i has 25 unit supply and sends 5 units to node j and 10 units to nodes k and l each. Note that the nodes in the network are in communication even if they are not their immediate adjacents since we assume that this communication is significant if the distance between any two nodes is at most L -hop. Accordingly, the supply sent through an edge $\{i, j\}$ contains 1) the signal from i to j and 2) signals from i to other connected nodes passing through j . To generate the demand representing the communication rate between nodes, we divide the flow amount in half and allocate each half to 1) and 2), respectively. We repeat this step for all nodes. Finally, each flow with different source and sink nodes corresponds to the demand values and belongs to a unique commodity of the instance. It is important to mention that there are two cases where we do not continue the flow and signals mentioned in 2) are lost. These cases are either node j is a dead-end or node j receives the signal at the L^{th} hop of the path. Still, only half of the entering signals contribute to the demand of node j .

Experiments are conducted in Python 3.10.5, using Spyder 5.3.1 programming language. We solve the mathematical models with Gurobi 9.5.2 (and the Gurobi Python library). In order to have a consistent comparison in computational experiments, we execute the codes on a MacBook Pro 16-inch (2023) computer with a 12-core processor running at 3.2 GHz and 16 GB RAM.

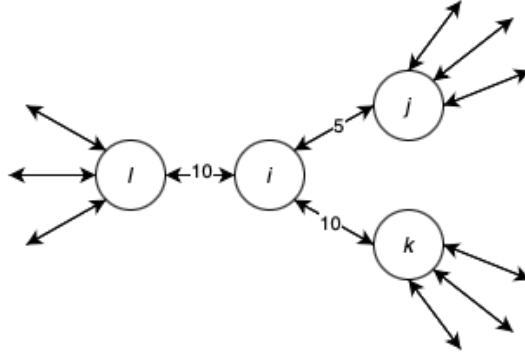


Figure 4.1: Example network for demand generation.

4.2 Computational Experiments

We conduct our computational experiments to track how the solution algorithms perform compared to the original formulations; furthermore, we analyze the change in solution times in both the solution methods and different model formulations, comparatively (single model approaches), under the different parameter settings.

4.2.1 Benders Decomposition Comparison

In the first experiment, we solve the global efficiency problem for 108 different instances using both Global Efficiency Model and Benders decomposition separately. Complete comparison is displayed in Tables 4.3, 4.4 and 4.5. Additionally, a comparison summary is presented in Table 4.2. In this summary table, we take the average of instances for each instance parameter individually. It is worth mentioning that the Benders decomposition algorithm is executed with a single CPU.

When we compare the total times of the two approaches, in terms of # of nodes, Global Efficiency Model is faster in small networks. However, the average difference is insignificant (≤ 1 sec), and as the network becomes larger, Benders decomposition becomes much faster. The average difference grows gradually

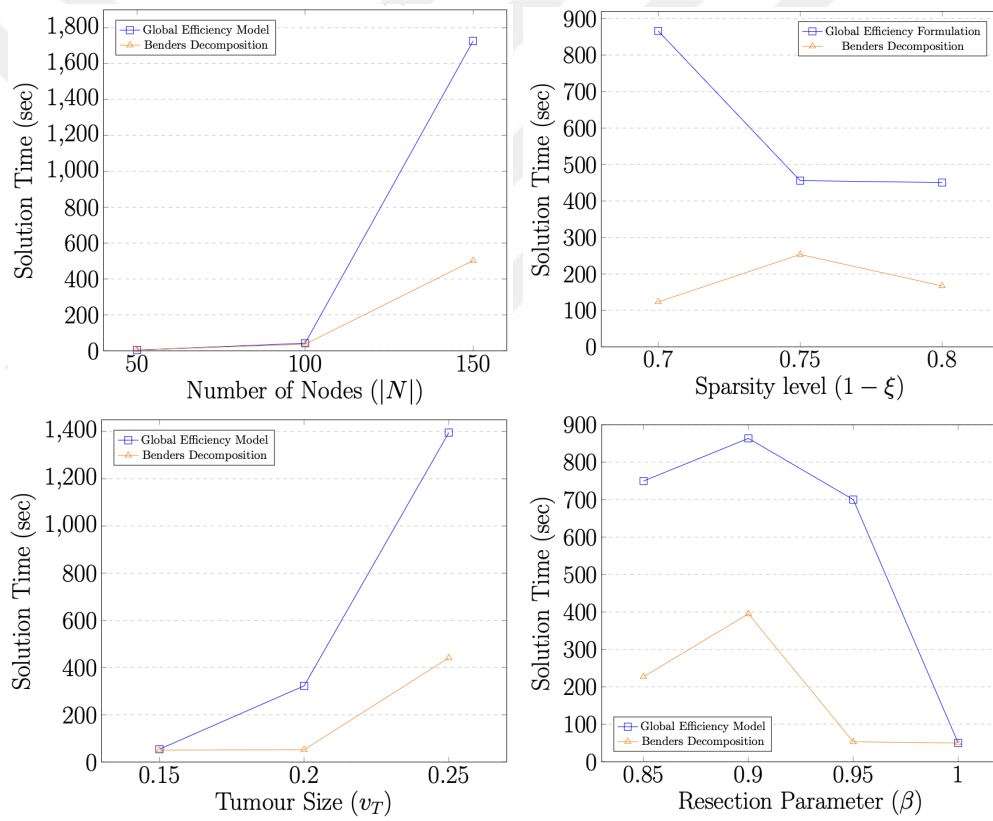


Figure 4.2: Average Solution time of Formulation 1 and Benders Decomposition for four instance parameters.

Table 4.2: Computational comparison summary between Global Efficiency Model & Benders Decomposition (Average)

Parameter	Fixed Value	Global Efficiency Model	Benders Decomposition			
		Solution Time (sec)	Construction Time (sec)	Solution Time (sec)	Total Time (sec)	Number of Iterations
$ N $	50	4.20	4.12	0.59	4.71	3.42
	100	42.07	32.48	4.07	36.55	4.50
	150	1726.34	109.71	392.92	502.63	3.94
ξ	0.2	450.57	48.80	118.36	167.16	3.89
	0.25	455.90	48.84	204.54	253.39	4.06
	0.3	866.15	48.68	74.67	123.35	3.92
v_T	0.15	54.39	48.65	1.68	50.33	2.39
	0.2	322.69	48.79	3.58	52.37	4.25
	0.25	1395.53	48.88	392.31	441.19	5.22
β	0.85	749.64	48.68	178.32	227.00	4.44
	0.9	863.56	48.78	346.20	394.98	4.81
	0.95	700.07	48.86	4.30	53.16	4.85
	1	50.21	48.77	1.29	50.05	1.70

where Global Efficiency Model takes 15% and 243% more time than Benders decomposition ($|N| = 100, 150$ respectively). If we further increase the network size to represent a brain with more detail, we expect a greater difference such that Global Efficiency Model fails to find a solution.

To continue with sparsity levels $(1 - \xi)$, Global Efficiency Model gets slower as the sparsity level decreases. Because when the nodes are connected by more edges in the network, resecting a node has more impact and this makes the decision process harder. On the other hand, in Benders decomposition, the decisions are made in different models (master problem and subproblems). Therefore, the network sparsity does not have a linear effect as the total time first increases and then decreases.

Tumour size has crucial significance on computation times. In both Global Efficiency Model and Benders decomposition, we realize a drastic rise in time as the tumour size increases. Still, Benders decomposition finds a solution faster in every configuration. Finally, they again have a similar direction of change in computation time based on the values of β . Although they start with similar computation times, when $\beta = 1$, Global Efficiency Model requires more time to find a solution in lower values of β .

We also analyze the computation time of Benders decomposition by considering model construction times and the model solution times. When we look at Tables 4.3, 4.4, and 4.5, the variance in computation time is insignificant for a fixed value of $|N|$. Hence the construction time depends on only $|N|$ among all the instance parameters, and the solution time is more sensitive to other instance parameters. Note that we solve dual sub-problems. Hence, we have an initial construction of models, then change the objective function in the problem. This implementation allows us to avoid these construction durations in each iteration. Finally, solving the master and subproblems in parallel processors, Benders decomposition will have promising results in much larger instances.

Table 4.3: Computational comparison between Global Efficiency Model & Benders Decomposition for ($|N| = 50$)

Instance ($ N , \xi, v_T, \beta$)	Global Efficiency Model	Benders Decomposition			
	Solution Time (sec)	Construction Time (sec)	Solution Time (sec)	Total Time (sec)	Number of Iterations
(50, 0.2, 0.15, 0.85)	4.15	4.14	0.23	4.37	1
(50, 0.2, 0.15, 0.9)	4.14	4.09	0.58	4.67	3
(50, 0.2, 0.15, 0.95)	4.12	4.03	0.62	4.65	4
(50, 0.2, 0.15, 1)	4.10	4.11	1.02	5.12	2
(50, 0.2, 0.2, 0.85)	4.27	4.07	0.67	4.74	5
(50, 0.2, 0.2, 0.9)	4.21	4.03	0.82	4.85	4
(50, 0.2, 0.2, 0.95)	4.18	3.98	0.59	4.57	4
(50, 0.2, 0.2, 1)	4.16	4.02	0.46	4.49	2
(50, 0.2, 0.25, 0.85)	4.17	4.05	0.97	5.02	5
(50, 0.2, 0.25, 0.9)	4.62	3.99	0.83	4.82	5
(50, 0.2, 0.25, 0.95)	4.19	4.11	0.89	5.00	6
(50, 0.2, 0.25, 1)	4.15	4.20	0.28	4.48	2
(50, 0.25, 0.15, 0.85)	4.13	4.13	0.23	4.36	1
(50, 0.25, 0.15, 0.9)	4.10	4.09	0.41	4.50	3
(50, 0.25, 0.15, 0.95)	4.14	4.18	0.39	4.57	3
(50, 0.25, 0.15, 1)	4.33	4.06	0.40	4.46	2
(50, 0.25, 0.2, 0.85)	4.18	4.06	0.62	4.68	4
(50, 0.25, 0.2, 0.9)	4.23	4.09	0.50	4.58	3
(50, 0.25, 0.2, 0.95)	4.17	4.19	0.53	4.71	4
(50, 0.25, 0.2, 1)	4.10	4.11	0.38	4.49	2
(50, 0.25, 0.25, 0.85)	4.19	4.14	1.03	5.16	6
(50, 0.25, 0.25, 0.9)	4.42	4.19	1.31	5.50	5
(50, 0.25, 0.25, 0.95)	4.22	4.17	1.18	5.35	7
(50, 0.25, 0.25, 1)	4.12	4.15	0.39	4.54	2
(50, 0.3, 0.15, 0.85)	4.19	4.11	0.22	4.33	1
(50, 0.3, 0.15, 0.9)	4.12	4.25	0.27	4.52	2
(50, 0.3, 0.15, 0.95)	4.27	4.30	0.41	4.70	2
(50, 0.3, 0.15, 1)	4.07	4.09	0.40	4.50	2
(50, 0.3, 0.2, 0.85)	4.21	4.08	0.62	4.71	5
(50, 0.3, 0.2, 0.9)	4.17	4.20	0.55	4.75	4
(50, 0.3, 0.2, 0.95)	4.35	4.21	0.75	4.96	5
(50, 0.3, 0.2, 1)	4.15	4.14	0.35	4.49	2
(50, 0.3, 0.25, 0.85)	4.26	4.15	0.66	4.81	4
(50, 0.3, 0.25, 0.9)	4.35	4.15	0.68	4.83	4
(50, 0.3, 0.25, 0.95)	4.26	4.25	0.64	4.90	5
(50, 0.3, 0.25, 1)	4.11	4.12	0.36	4.47	2

Table 4.4: Computational comparison between Global Efficiency Model & Benders Decomposition for ($|N| = 100$)

Instance ($ N , \xi, v_T, \beta$)	Global Efficiency Model	Benders Decomposition			
	Solution Time (sec)	Construction Time (sec)	Solution Time (sec)	Total Time (sec)	Number of Iterations
(100, 0.2, 0.15, 0.85)	32.84	31.96	1.01	32.97	1
(100, 0.2, 0.15, 0.9)	34.97	32.33	2.12	34.46	4
(100, 0.2, 0.15, 0.95)	35.21	32.38	2.06	34.44	4
(100, 0.2, 0.15, 1)	33.35	32.60	1.21	33.81	2
(100, 0.2, 0.2, 0.85)	37.12	32.56	2.39	34.94	5
(100, 0.2, 0.2, 0.9)	43.69	32.43	3.37	35.80	5
(100, 0.2, 0.2, 0.95)	38.56	32.61	2.28	34.89	5
(100, 0.2, 0.2, 1)	33.78	32.63	1.18	33.81	2
(100, 0.2, 0.25, 0.85)	90.51	32.54	12.90	45.44	7
(100, 0.2, 0.25, 0.9)	97.07	32.68	12.34	45.02	7
(100, 0.2, 0.25, 0.95)	57.20	31.97	3.72	35.69	7
(100, 0.2, 0.25, 1)	33.82	32.42	1.14	33.56	2
(100, 0.25, 0.15, 0.85)	33.57	32.43	0.94	33.37	1
(100, 0.25, 0.15, 0.9)	38.07	32.23	2.46	34.69	5
(100, 0.25, 0.15, 0.95)	36.31	32.31	2.22	34.53	4
(100, 0.25, 0.15, 1)	33.68	32.48	1.12	33.60	2
(100, 0.25, 0.2, 0.85)	43.58	32.70	3.83	36.53	8
(100, 0.25, 0.2, 0.9)	57.76	32.71	3.72	36.44	6
(100, 0.25, 0.2, 0.95)	54.97	32.63	2.34	34.96	5
(100, 0.25, 0.2, 1)	33.55	32.61	1.15	33.76	2
(100, 0.25, 0.25, 0.85)	38.23	32.70	24.46	57.17	9
(100, 0.25, 0.25, 0.9)	46.32	32.41	18.95	51.36	8
(100, 0.25, 0.25, 0.95)	38.49	32.55	4.19	36.74	7
(100, 0.25, 0.25, 1)	32.80	32.54	1.26	33.80	2
(100, 0.3, 0.15, 0.85)	33.23	32.22	0.90	33.11	1
(100, 0.3, 0.15, 0.9)	35.08	32.47	1.68	34.15	3
(100, 0.3, 0.15, 0.95)	34.74	32.45	2.11	34.56	4
(100, 0.3, 0.15, 1)	33.58	32.45	0.91	33.37	1
(100, 0.3, 0.2, 0.85)	40.67	32.52	2.95	35.47	6
(100, 0.3, 0.2, 0.9)	51.64	32.32	3.56	35.88	6
(100, 0.3, 0.2, 0.95)	41.24	32.63	2.95	35.58	6
(100, 0.3, 0.2, 1)	33.68	32.58	0.88	33.46	1
(100, 0.3, 0.25, 0.85)	38.99	32.68	6.62	39.31	8
(100, 0.3, 0.25, 0.9)	43.28	32.43	7.16	39.59	8
(100, 0.3, 0.25, 0.95)	39.11	32.67	3.04	35.71	6
(100, 0.3, 0.25, 1)	33.94	32.49	1.23	33.72	2

Table 4.5: Computational comparison between Global Efficiency Model & Benders Decomposition for ($|N| = 150$)

Instance ($ N , \xi, v_T, \beta$)	Global Efficiency Model	Benders Decomposition			
	Solution Time (sec)	Construction Time (sec)	Solution Time (sec)	Total Time (sec)	Number of Iterations
(150, 0.2, 0.15, 0.85)	113.88	109.86	1.99	111.85	1
(150, 0.2, 0.15, 0.9)	119.12	109.76	3.58	113.35	3
(150, 0.2, 0.15, 0.95)	120.87	108.77	3.41	112.18	3
(150, 0.2, 0.15, 1)	112.30	109.46	2.65	112.11	2
(150, 0.2, 0.2, 0.85)	270.00	109.79	10.99	120.78	6
(150, 0.2, 0.2, 0.9)	543.14	110.16	12.69	122.84	4
(150, 0.2, 0.2, 0.95)	191.16	109.93	6.08	116.01	5
(150, 0.2, 0.2, 1)	113.03	109.51	2.59	112.09	2
(150, 0.2, 0.25, 0.85)	5211.98	109.56	1822.87	1932.43	6
(150, 0.2, 0.25, 0.9)	5461.22	109.65	2325.56	2435.21	6
(150, 0.2, 0.25, 0.95)	3231.65	110.10	12.31	122.41	6
(150, 0.2, 0.25, 1)	113.61	112.16	2.64	114.80	2
(150, 0.25, 0.15, 0.85)	119.59	109.52	2.00	111.52	1
(150, 0.25, 0.15, 0.9)	137.82	109.78	4.31	114.09	4
(150, 0.25, 0.15, 0.95)	128.84	109.40	3.51	112.92	3
(150, 0.25, 0.15, 1)	112.15	109.28	2.11	111.39	1
(150, 0.25, 0.2, 0.85)	484.48	110.12	7.21	117.33	6
(150, 0.25, 0.2, 0.9)	948.43	109.91	12.65	122.56	5
(150, 0.25, 0.2, 0.95)	257.47	110.69	5.49	116.18	5
(150, 0.25, 0.2, 1)	112.95	109.22	1.97	111.18	1
(150, 0.25, 0.25, 0.85)	5040.34	109.85	2172.50	2282.35	6
(150, 0.25, 0.25, 0.9)	5655.11	109.09	5066.40	5175.49	6
(150, 0.25, 0.25, 0.95)	2764.99	110.76	9.26	120.02	6
(150, 0.25, 0.25, 1)	112.40	110.87	2.21	113.07	1
(150, 0.3, 0.15, 0.85)	113.72	109.24	2.22	111.46	1
(150, 0.3, 0.15, 0.9)	154.58	109.52	4.44	113.96	4
(150, 0.3, 0.15, 0.95)	147.53	109.93	4.19	114.12	4
(150, 0.3, 0.15, 1)	113.21	108.88	2.25	111.13	1
(150, 0.3, 0.2, 0.85)	657.88	108.72	7.66	116.38	7
(150, 0.3, 0.2, 0.9)	4268.56	109.02	15.86	124.89	5
(150, 0.3, 0.2, 0.95)	3095.68	109.35	6.08	115.43	5
(150, 0.3, 0.2, 1)	113.46	109.96	2.24	112.19	1
(150, 0.3, 0.25, 0.85)	7801.91	108.48	725.88	834.36	8
(150, 0.3, 0.25, 0.9)	5541.96	111.13	1840.53	1951.66	8
(150, 0.3, 0.25, 0.95)	8549.93	110.56	34.86	145.42	6
(150, 0.3, 0.25, 1)	113.20	107.54	2.01	109.56	1

4.2.2 Column Generation Comparison

We compare the Column Generation algorithm with CFP, starting with smaller-sized instances where $|N|$ is 10 and increment by 10 up to 50, which is the lowest node size in the original instance set. Table 4.6 shows the solution time of these five instances. We observe that even in small instances, column generation solution time increases exponentially, as can be seen in Figure 4.3. This increase is also tractable at the number of iterations; for example, the last instance finds a solution in 387 iterations while the first instance finds it in 18 iterations. We interpret this result as $|N|$ increases $\#$ of paths overloads the master problem, which can also be observed at $\#$ of iterations and $\#$ of generated paths. The solution time increases with higher acceleration than $\#$ of iterations, and extreme solution times are realized when the algorithm generates too many paths and sub-problems fail to find effective columns. Thus, we do not experiment with the original instance set.

Table 4.6: Computational comparison between CFP & Column Generation (time in seconds)

	$ N $	10	20	30	40	50
CFP	Time	0.0111	0.0639	0.0639	0.6928	1.2764
Column Generation	Time	0.4697	21.9887	180.1156	966.4379	4,830.5757
	$\#$ of paths	181	1147	2485	4244	10156
	$\#$ of iterations	19	47	121	225	398
Column Generation (no resection)	Time	0.3584	18.7830	159.6734	926.2208	4,619.6253
	$\#$ of paths	184	1120	2428	4265	6539
	$\#$ of iterations	18	45	115	227	387

Computational Comparison Between CFP & Column Generation

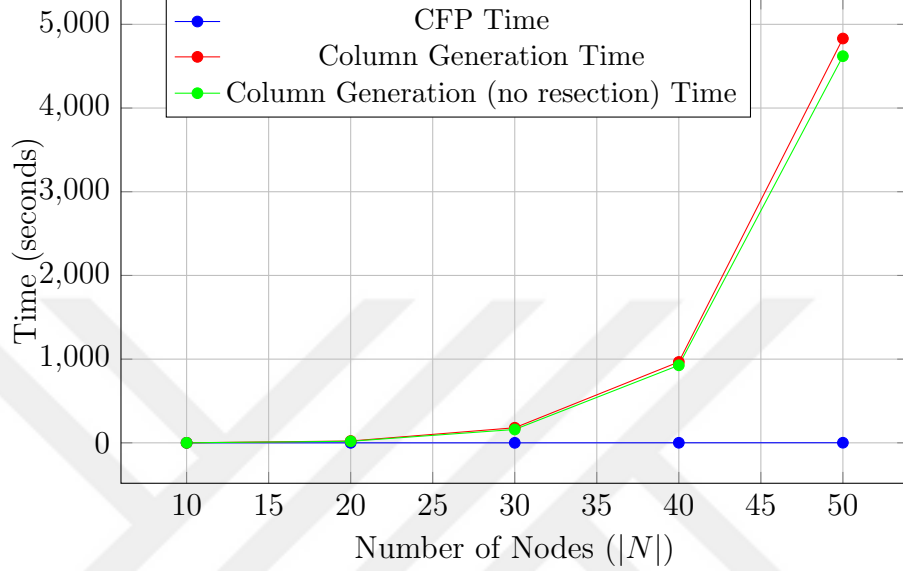


Figure 4.3: Computational comparison between CFP & Column Generation (time in seconds)

4.3 Solution Analysis

In this section we perform solution analysis for global efficiency experiments and study the solution behavior. The optimal global efficiency levels for our instance set are displayed in Table 4.7. According to this table, global efficiency does not have a clear relation with tumour size, which can be seen clearly in Figure 4.4. On the other hand, if we resect more tumourous nodes, the global efficiency decreases. Note that, this change is an optimal decision; therefore, a decision without an optimization model may return much lower global efficiency.

Similarly, we check which nodes are resected and present in Table 4.8, 4.9, and 4.10. We see that when $|N|$ is 50, as we move from $\beta = 1$ to $\beta = 0.85$, the model generally chooses the same nodes and adds new ones to resect; but when $|N|$ is larger, in the optimal solutions, we observe that resection decisions are changing.

In addition to analyzing the global efficiency model outputs, we experiment with the performance of these solutions in terms of other measures we mentioned.

Table 4.7: Solution analysis of objectives

	$ N $	50			100			150		
v_T	ξ β	0.2	0.25	0.3	0.2	0.25	0.3	0.2	0.25	0.3
0.15	0.85	0.427	0.456	0.478	0.438	0.451	0.465	0.445	0.464	0.482
	0.9	0.471	0.508	0.528	0.492	0.508	0.525	0.490	0.509	0.528
	0.95	0.514	0.557	0.577	0.548	0.567	0.585	0.542	0.563	0.584
	1	0.579	0.628	0.650	0.604	0.627	0.647	0.603	0.626	0.650
0.2	0.85	0.410	0.439	0.461	0.439	0.452	0.468	0.433	0.451	0.468
	0.9	0.474	0.510	0.530	0.492	0.508	0.526	0.490	0.510	0.529
	0.95	0.516	0.557	0.577	0.548	0.567	0.586	0.542	0.563	0.585
	1	0.579	0.628	0.650	0.604	0.627	0.647	0.603	0.626	0.650
0.25	0.85	0.414	0.442	0.461	0.440	0.453	0.469	0.434	0.452	0.469
	0.9	0.475	0.510	0.530	0.493	0.509	0.526	0.491	0.510	0.529
	0.95	0.516	0.557	0.577	0.548	0.567	0.586	0.542	0.563	0.585
	1	0.579	0.628	0.650	0.604	0.627	0.647	0.603	0.626	0.650

For this analysis, we input the optimal resection decision of the global efficiency model to CFP and CP; and track the objective function values. Table 4.11 displays the values of each measure we suggest under given resection decisions. This experiment is conducted for a new instance subset where $|N| = 50$, $\xi = 0.2$ which is different from the previous graphs, to get a comprehensive experimental result of how the models comparatively work. The experiment is furthered by solving the CFP and CP models individually and implementing their solutions to the other models separately as shown in 4.12 and 4.13. The aim of this experiment is to observe how the optimal values behave on different models and see if these models are interchangeable in terms of objectives and solutions.

The tables do not show a strict trend for optimal values of $z_{\min}$ and ϕ measures with changing v_T or β . When the tumor size is small ($v_T = 0.15$), the optimal resection decision for $z_{\min}$ is also optimal for the ϕ measure. Similar results occur in the opposite direction, except where $\beta = 0.9$. This indicates alternative optimal solutions for ϕ , with only some also being optimal for $z_{\min}$. For larger tumor sizes, different values are observed. There is no significant relation between global efficiency and the other two objectives, as the optimal decision for global efficiency does not yield the best $z_{\min}$ or ϕ . This is also true in the opposite direction. This result is expected since global efficiency neglects white matter thickness.

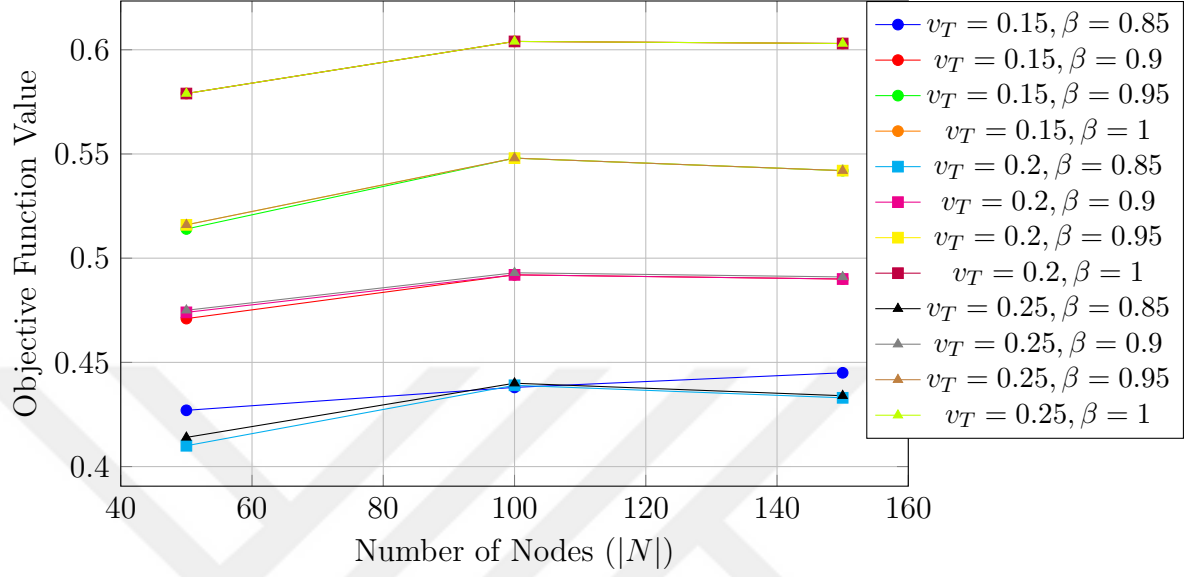


Figure 4.4: Global Efficiency Objective Function Values for Different β and v_T Values

However, it can be concluded that the CFP and CP models are giving parallel results. Lastly, Analyzing global efficiency individually shows an increasing trend, directly correlated to β values, also found in the initial instance set (see 4.7).

Table 4.8: Resection output in global efficiency model for ($|N| = 50$)

Instance ($ N , \xi, v_T, \beta$)	Resection Decision (1: resected, 0: not resected)
(50, 0.2, 0.15, 0.85)	[1 1 1 1 1 1 1]
(50, 0.2, 0.15, 0.9)	[1 1 1 0 1 1 0]
(50, 0.2, 0.15, 0.95)	[1 0 1 0 1 0 0]
(50, 0.2, 0.15, 1)	[0 0 0 0 0 0 0]
(50, 0.2, 0.2, 0.85)	[1 1 1 1 0 1 0 1 1 1]
(50, 0.2, 0.2, 0.9)	[1 0 1 0 1 1 0 0 0 1]
(50, 0.2, 0.2, 0.95)	[1 0 0 0 0 1 0 0 0 1]
(50, 0.2, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0]
(50, 0.2, 0.25, 0.85)	[1 0 1 1 1 0 0 1 0 1 1 1]
(50, 0.2, 0.25, 0.9)	[1 0 1 0 1 0 0 0 0 1 0 1]
(50, 0.2, 0.25, 0.95)	[1 0 0 0 0 1 0 0 0 0 0 1]
(50, 0.2, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0]
(50, 0.25, 0.15, 0.85)	[1 1 1 1 1 1 1]
(50, 0.25, 0.15, 0.9)	[0 1 1 0 1 1 1]
(50, 0.25, 0.15, 0.95)	[0 1 0 0 1 1 0]
(50, 0.25, 0.15, 1)	[0 0 0 0 0 0 0]
(50, 0.25, 0.2, 0.85)	[0 1 1 0 1 1 1 1 1 1]
(50, 0.25, 0.2, 0.9)	[1 1 1 0 1 0 1 0 0 0]
(50, 0.25, 0.2, 0.95)	[0 1 1 0 0 0 0 0 0 1]
(50, 0.25, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0]
(50, 0.25, 0.25, 0.85)	[0 1 1 0 1 1 0 1 1 1 0 1]
(50, 0.25, 0.25, 0.9)	[0 1 0 0 1 1 0 1 0 0 0 1]
(50, 0.25, 0.25, 0.95)	[0 1 1 0 1 0 0 0 0 0 0 0]
(50, 0.25, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0]
(50, 0.3, 0.15, 0.85)	[1 1 1 1 1 1 1]
(50, 0.3, 0.15, 0.9)	[1 0 1 0 1 1 1]
(50, 0.3, 0.15, 0.95)	[0 1 0 0 1 0 1]
(50, 0.3, 0.15, 1)	[0 0 0 0 0 0 0]
(50, 0.3, 0.2, 0.85)	[1 1 1 0 1 1 1 1 0 1]
(50, 0.3, 0.2, 0.9)	[1 1 0 0 0 1 1 0 0 1]
(50, 0.3, 0.2, 0.95)	[1 0 0 0 0 1 1 0 0 0]
(50, 0.3, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0]
(50, 0.3, 0.25, 0.85)	[0 1 1 0 0 1 1 1 1 1 0 1]
(50, 0.3, 0.25, 0.9)	[1 0 1 0 0 0 1 0 1 1 0 0]
(50, 0.3, 0.25, 0.95)	[0 0 1 0 0 0 1 0 0 1 0 0]
(50, 0.3, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0]

Table 4.9: Resection output in global efficiency model for ($|N| = 100$)

Instance ($ N , \xi, v_T, \beta$)	Resection Decision (1: resected, 0: not resected)
(100, 0.2, 0.15, 0.85)	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
(100, 0.2, 0.15, 0.9)	[1 0 0 1 1 1 0 0 0 1 1 1 1 1 1]
(100, 0.2, 0.15, 0.95)	[0 0 1 0 0 0 0 1 1 1 0 0 0 0 1]
(100, 0.2, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.2, 0.2, 0.85)	[1 1 1 1 1 0 1 0 1 1 0 1 1 1 0 1 1]
(100, 0.2, 0.2, 0.9)	[1 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 0]
(100, 0.2, 0.2, 0.95)	[0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0]
(100, 0.2, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.2, 0.25, 0.85)	[1 1 0 0 1 1 0 1 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 1 1]
(100, 0.2, 0.25, 0.9)	[1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1]
(100, 0.2, 0.25, 0.95)	[0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1]
(100, 0.2, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.25, 0.15, 0.85)	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
(100, 0.25, 0.15, 0.9)	[1 1 0 1 1 1 0 0 0 1 1 1 1 0 1]
(100, 0.25, 0.15, 0.95)	[0 0 0 0 1 0 0 1 0 0 1 1 1 0 0]
(100, 0.25, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.25, 0.2, 0.85)	[0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0]
(100, 0.25, 0.2, 0.9)	[0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0]
(100, 0.25, 0.2, 0.95)	[0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1]
(100, 0.25, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.25, 0.25, 0.85)	[0 0 1 1 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 1 1 1]
(100, 0.25, 0.25, 0.9)	[0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1]
(100, 0.25, 0.25, 0.95)	[0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1]
(100, 0.25, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.3, 0.15, 0.85)	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
(100, 0.3, 0.15, 0.9)	[1 1 1 0 0 1 1 1 0 0 1 0 1 1 1]
(100, 0.3, 0.15, 0.95)	[1 0 1 0 0 0 1 0 1 0 0 0 1 0 0]
(100, 0.3, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.3, 0.2, 0.85)	[1 0 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1]
(100, 0.3, 0.2, 0.9)	[0 1 1 1 0 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1]
(100, 0.3, 0.2, 0.95)	[1 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0]
(100, 0.3, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(100, 0.3, 0.25, 0.85)	[1 0 1 0 1 0 1 0 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0]
(100, 0.3, 0.25, 0.9)	[1 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1]
(100, 0.3, 0.25, 0.95)	[1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
(100, 0.3, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Table 4.10: Resection output in global efficiency model for ($|N| = 150$)

Instance ($ N , \xi, v_T, \beta$)	Resection Decision (1: resected, 0: not resected)
(150, 0.2, 0.15, 0.85)	[1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
(150, 0.2, 0.15, 0.9)	[1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0]
(150, 0.2, 0.15, 0.95)	[0 1 1 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0 0]
(150, 0.2, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.2, 0.2, 0.85)	[1 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1]
(150, 0.2, 0.2, 0.9)	[0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 1 0 0]
(150, 0.2, 0.2, 0.95)	[0 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0]
(150, 0.2, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.2, 0.25, 0.85)	[0 1 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 0 1 0 1 0 1 1 1]
(150, 0.2, 0.25, 0.9)	[0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 1 0 0]
(150, 0.2, 0.25, 0.95)	[0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0]
(150, 0.2, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.25, 0.15, 0.85)	[0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
(150, 0.25, 0.15, 0.9)	[0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1]
(150, 0.25, 0.15, 0.95)	[0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 0 0]
(150, 0.25, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.25, 0.2, 0.85)	[0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1]
(150, 0.25, 0.2, 0.9)	[0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1]
(150, 0.25, 0.2, 0.95)	[0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0]
(150, 0.25, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.25, 0.25, 0.85)	[0 1 0 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 0]
(150, 0.25, 0.25, 0.9)	[0 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0]
(150, 0.25, 0.25, 0.95)	[1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0]
(150, 0.25, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.3, 0.15, 0.85)	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1]
(150, 0.3, 0.15, 0.9)	[1 1 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 0 0 0 1 1]
(150, 0.3, 0.15, 0.95)	[1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 1 0]
(150, 0.3, 0.15, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.3, 0.2, 0.85)	[1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1]
(150, 0.3, 0.2, 0.9)	[1 0 0 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 0 0 0 0]
(150, 0.3, 0.2, 0.95)	[1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0]
(150, 0.3, 0.2, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(150, 0.3, 0.25, 0.85)	[1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 1 1 1 1 1]
(150, 0.3, 0.25, 0.9)	[1 1 0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1]
(150, 0.3, 0.25, 0.95)	[1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1]
(150, 0.3, 0.25, 1)	[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Table 4.11: Solution analysis for global efficiency results($|N| = 50, \xi = 0.20$)

v_T	β	Global Efficiency	$z_{\min}$	ϕ
0.15	0.85	0.419	39.931%	3.495
	0.9	0.462	42.781%	3.389
	0.95	0.506	43.713%	3.357
	1	0.569	45.612%	3.341
0.2	0.85	0.402	37.004%	3.690
	0.9	0.464	41.307%	3.495
	0.95	0.506	35.890%	3.572
	1	0.569	45.612%	3.341

Table 4.12: Solution analysis for $z_{\min}$ results($|N| = 50, \xi = 0.20$)

v_T	β	Global Efficiency	$z_{\min}$	ϕ
0.15	0.85	0.419	39.931%	3.495
	0.9	0.462	47.321%	3.361
	0.95	0.504	45.413%	3.361
	1	0.569	45.612%	3.341
0.2	0.85	0.393	47.300%	3.56
	0.9	0.442	46.300%	3.56
	0.95	0.504	43.800%	3.717
	1	0.569	45.612%	3.341

Table 4.13: Solution analysis for ϕ results($|N| = 50, \xi = 0.20$)

v_T	β	Global Efficiency	$z_{\min}$	ϕ
0.15	0.85	0.419	39.931%	3.495
	0.9	0.503	46.200%	3.361
	0.95	0.504	45.413%	3.361
	1	0.569	45.612%	3.341
0.2	0.85	0.400	40.800%	3.452
	0.9	0.439	41.600%	3.560
	0.95	0.399	42.300%	3.560
	1	0.569	45.612%	3.341

Chapter 5

Conclusion

In this thesis, we aim to propose a systematic decision-making tool for neurosurgeons for resecting brain tumours from a patient. Since each patient is a unique case, we answer the question, "Which tumourous parts should be selected in order to cause minimal side effects on the brain?" which refers to a specific action rather than a procedural one. To be more sensitive to the case, we choose to quantify the effects of the operation on the brain by using three different measures: global efficiency, demand satisfaction rate in concurrent flows, and congestion. We develop different models for each measure. Our model for global efficiency is a critical node based model, and we suggest a reformulation and construct a Benders decomposition algorithm. For the other measures, we used flow-based modeling approaches and formulate the Concurrent Flow Problem (CFP) and the Congestion Problem (CP) models. We implement a column generation algorithm to solve CFP under given resection decisions. We test the computational performance of the algorithms using a diverse instance set and finally analyze the solution outputs.

We observed that the Benders decomposition algorithm performs much better than solving the Global Efficiency Model. Furthermore, the difference gets better in larger instances such that the Bender decomposition gives promises to answer the cases closer to representing the brain network. Although the Global

Efficiency Model fails in larger instances, it makes parallel computing possible, which contributes a pivotal role in the computational challenges. On the other hand, we found that column generation is not applicable to CFP. We saw that optimal global efficiency decreases as the number of resected nodes increases. Finally, for a given resection decision, both CFP and CP behaves similarly to the changes in problem parameters.

In future studies, in terms of computational improvement, we can add modifications like eliminating columns that are not used for a certain number of iterations or finding different methods to generate paths. Another extension could be using a more detailed global efficiency function that considers the edge capacity in addition to hop length. One additional study can be an implementation of adjustments on formulations by adding node importance to nodes that represent the brain part having more crucial tasks. Instead of maximizing the $z_{\min}$ of the network, we can maximize the weighted average where node volumes define these weights.

Bibliography

- [1] F. Gaillard, “Normal brain (mri). case study,” *Radiopaedia.org*, 7 2015.
- [2] N. Pouratian, A. Asthagiri, J. Jagannathan, M. Shaffrey, and D. Schiff, “Surgery insight: The role of surgery in the management of low-grade gliomas,” *Nature clinical practice. Neurology*, vol. 3, pp. 628–39, 12 2007.
- [3] R. Nieuwenhuys, J. Voogd, and C. van Huijzen, *The Human Central Nervous System*. Springer, 4th ed., 2007.
- [4] Cancer Research UK, “Types of brain tumours,” n.d. Accessed: 2024-06-28.
- [5] G. B. of Disease Collaborative Network, “Global burden of brain and other central nervous system cancer: Findings from the global burden of disease study 2019,” *Archives of Public Health*, vol. 80, no. 1, pp. 1–18, 2022.
- [6] American Brain Tumor Association, “Brain tumor education,” n.d. Accessed: 2024-06-28.
- [7] MD Anderson Cancer Center, “Brain tumor treatment,” n.d. Accessed: 2024-06-28.
- [8] BrainLine, “Brain imaging: What are the different types?,” 2022. Accessed: 2024-06-28.
- [9] A. W. Toga and J. C. Mazziotta, *Brain Mapping: The Methods*. San Diego: Academic Press, 1998.

- [10] T. E. Conturo, N. F. Lori, T. S. Cull, E. Akbudak, A. Z. Snyder, J. S. Shimony, R. C. McKinstry, H. Burton, and M. E. Raichle, “Tracking neuronal fiber pathways in the living human brain,” *Proceedings of the National Academy of Sciences*, vol. 96, no. 18, pp. 10422–10427, 1999.
- [11] R. Kötter and E. Wanke, “Mapping brains without coordinates,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 751–766, 2005.
- [12] O. Sporns, G. Tononi, and R. Kötter, “The human connectome: A structural description of the human brain,” *PLoS Computational Biology*, vol. 1, no. 4, p. e42, 2005.
- [13] K. Stephan, L. Kamper, A. Bozkurt, G. Burns, M. Young, and R. Kötter, “Advanced database methodology for the collation of connectivity data on the macaque brain (cocomac),” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 356, no. 1412, pp. 1159–1186, 2001.
- [14] V. Wedeen, P. Hagmann, W. Tseng, T. Reese, and R. Weisskoff, “Mapping fiber orientation spectra in cerebral white matter with fourier-transform diffusion mri,” *Proceedings of the International Society for Magnetic Resonance in Medicine*, vol. 13, p. 82, 2005.
- [15] O. Sporns, C. J. Honey, and R. Kötter, “Identification and classification of hubs in brain networks,” *PLoS ONE*, vol. 2, no. 10, p. e1049, 2007.
- [16] C. J. Honey and O. Sporns, “Dynamical consequences of lesions in cortical networks,” *Hum Brain Mapp*, vol. 29, no. 7, pp. 802–809, 2008.
- [17] H. Johansen-Berg and T. E. Behrens, *Diffusion MRI: From Quantitative Measurement to In vivo Neuroanatomy*. San Diego, CA: Elsevier Academic Press, 2005.
- [18] M. F. Glasser, T. S. Coalson, E. C. Robinson, C. D. Hacker, J. Harwell, E. Yacoub, K. Ugurbil, J. Andersson, C. F. Beckmann, M. Jenkinson, *et al.*, “A multi-modal parcellation of human cerebral cortex,” *Nature*, vol. 536, pp. 171–178, 2016.

- [19] M. Schirner, S. Rothmeier, V. K. Jirsa, A. R. McIntosh, and P. Ritter, “An automated pipeline for constructing personalized virtual brains from multimodal neuroimaging data,” *NeuroImage*, vol. 117, pp. 343–357, 2015.
- [20] H. Aerts, M. Schirner, B. Jeurissen, D. Van Roost, R. Achten, P. Ritter, and D. Marinazzo, “Modeling brain dynamics in brain tumor patients using the virtual brain,” *eNeuro*, vol. 5, no. 3, pp. ENEURO.0083–18.2018, 2018.
- [21] E. Bullmore and O. Sporns, “The economy of brain network organization,” *Nature Reviews Neuroscience*, vol. 13, no. 5, pp. 336–349, 2012.
- [22] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. E. Hulshoff Pol, “Efficient brain networks and intelligence,” *Trends in Cognitive Sciences*, vol. 13, no. 6, pp. 214–221, 2009.
- [23] A. Bessadok and I. Rekik, “Graph neural networks in network neuroscience,” *Frontiers in Neuroscience*, vol. 15, 2021.
- [24] S. Meesters, M. Landers, G. Rutten, and L. Florack, “Subject-specific automatic reconstruction of white matter tracts,” *Journal of Digital Imaging*, vol. 36, no. 6, pp. 2648–2661, 2023.
- [25] S. P. K. D. Mukherkjee, D. *et al.*, “Brain tumor image generation using an aggregation of gan models with style transfer,” *Scientific Reports*, vol. 12, p. 9141, 2022.
- [26] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, “nnunet: Self-adapting framework for u-net-based medical image segmentation,” *arXiv preprint arXiv:2011.00848*, 2020.
- [27] D. Liu, J. Chen, X. Hu, K. Yang, Y. Liu, G. Hu, H. Ge, W. Zhang, and H. Liu, “Imaging-genomics in glioblastoma: Combining molecular and imaging signatures,” *Frontiers in Oncology*, vol. 11, 2021.
- [28] A. Arulselvan, C. W. Commander, L. Eleftheriadou, and P. M. Pardalos, “Detecting critical nodes in sparse graphs,” *Computers & Operations Research*, vol. 36, no. 7, pp. 2193–2200, 2009.

- [29] D. Granata, G. Steeger, and S. Rebennack, “Network interdiction via a critical disruption path: Branch-and-price algorithms,” *Computers & Operations Research*, vol. 40, no. 11, pp. 2689–2702, 2013.
- [30] A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao, “Critical nodes for distance-based connectivity and related problems in graphs,” *Networks*, vol. 66, no. 3, pp. 170–195, 2015.
- [31] A. T. Murray, T. C. Matisziw, and T. H. Grubeshic, “Critical network infrastructure analysis: interdiction and system flow,” *Journal of Geographical Systems*, vol. 9, no. 2, pp. 103–117, 2007.
- [32] M. Ventresca, K. R. Harrison, and B. M. Ombuki-Berman, “The bi-objective critical node detection problem,” *Applied Soft Computing*, vol. 265, pp. 895–908, 2018.
- [33] E.-Y. Yu, Y. Fu, X. Chen, M. Xie, and D.-B. Chen, “Identifying critical nodes in temporal networks by network embedding,” *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.
- [34] J. Zhong, F. Zhang, and Z. Li, “Identification of vital nodes in complex network via belief propagation and node reinsertion,” *IEEE Access*, vol. 6, pp. 29200–29210, 2018.
- [35] M. Lalou, M. A. Tahraoui, and H. Kheddouci, “The critical node detection problem in networks: A survey,” *Computer Science Review*, vol. 28, pp. 92–117, 2018.
- [36] F. Shahrokhi and D. W. Matula, “The maximum concurrent flow problem,” *Journal of the ACM*, vol. 37, no. 2, pp. 318–334, 1990.
- [37] L. Gouveia, “Multicommodity flow models for spanning trees with hop constraints,” *European Journal of Operational Research*, vol. 95, no. 1, pp. 178–190, 1996.
- [38] W. Dai, X. Sun, and S. Wandelt, “Finding feasible solutions for multi-commodity flow problems,” in *Chinese Control Conference (CCC)*, pp. 5925–5930, 2016.

- [39] W. Dai, J. Zhang, and X. Sun, “On solving multi-commodity flow problems: An experimental evaluation,” *Chinese Journal of Aeronautics*, vol. 30, no. 2, pp. 519–529, 2017.
- [40] S. Moradi, A. Raith, and M. Ehrgott, “A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem,” *European Journal of Operational Research*, vol. 244, no. 2, pp. 369–378, 2015.
- [41] P. Liu, “A combinatorial algorithm for the multi-commodity flow problem,” *arXiv preprint arXiv:1904.09397*, 2019.
- [42] K. Salimifard and S. Bigharaz, “The multicommodity network flow problem: state of the art classification, applications, and solution methods,” *Operational Research*, vol. 22, no. 1, pp. 1–47, 2022.
- [43] H. Racke, “Minimizing congestion in general networks,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 43–52, 2002.
- [44] H. Räcke, “Optimal hierarchical decompositions for congestion minimization in networks,” in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08*, (New York, NY, USA), p. 255–264, Association for Computing Machinery, 2008.
- [45] R. Banner and A. Orda, “Multipath routing algorithms for congestion minimization,” in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems* (R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. P. Black, eds.), (Berlin, Heidelberg), pp. 536–548, Springer Berlin Heidelberg, 2005.
- [46] K. Motomura, F. Ohka, K. Aoki, and R. Saito, “Supratotal resection of gliomas with awake brain mapping: Maximal tumor resection preserving motor, language, and neurocognitive functions,” *Frontiers in Neurology*, vol. 13, p. 874826, 2022.

- [47] M. Rossi, F. Ambrogi, L. Gay, M. Gallucci, M. Conti Nibali, A. Leonetti, G. Puglisi, T. Sciortino, H. Howells, M. Riva, *et al.*, “Is supratotal resection achievable in low-grade gliomas? feasibility, putative factors, safety, and functional outcome,” *Journal of Neurosurgery*, vol. 132, no. 6, pp. 1692–1705, 2019.
- [48] T. Khaniyev, K. Cakiroglu, A. S. Bozkurt, O. Prokopyev, and S. Hanalioglu, “Optimal brain tumor resection,” 2024. To be submitted to POMS. (Work in Progress).
- [49] S. O. Mišić, B. and A. McIntosh, “Communication efficiency and congestion of signal traffic in large-scale brain networks,” *PLOS Computational Biology*, vol. 10, no. 1, p. e1003427, 2014.
- [50] Q. Hu and X.-D. Zhang, “Fundamental patterns of signal propagation in complex networks,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, no. 1, p. 013149, 2024.