OPTIMIZING QUANTUM-INSPIRED COMPUTATIONAL TECHNIQUES FOR QUADRATIC UNCONSTRAINED BINARY OPTIMIZATION PROBLEM SOLVING

MR. PANITHI SUWANNO
MR. MUHAMMUD BINHAR

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2023

Optimizing Quantum-inspired Computational Techniques for Quadratic Unconstrained Binary Optimization Problem Solving

Mr. Panithi Suwanno
Mr. Muhammud Binhar

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2023

Project Committee

[illegible] .... (Asst. Prof. Rajchawit Sarochawikasit)	Project Advisor
[illegible] .... (Unchalisa Taetragool, Ph.D.)	Committee Member
[illegible] .... (Assoc. Prof. Naruemon Wattanapongsakorn, Ph.D.)	Committee Member
[illegible] .... (Assoc. Prof. Natasha Dejdumrong, D.Tech.Sci.)	Committee Member

Project Title	Optimizing Quantum-inspired Computational Techniques for Quadratic Unconstrained Binary Optimization Problem Solving
Credits	3
Member(s)	Mr. Panithi Suwanno Mr. Muhammud Binhar
Project Advisor	Asst. Prof. Rajchawit Sarochawikasit
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2023

Abstract

This research uses quantum-inspired computing techniques to enhance computational efficiency for solving Quadratic Unconstrained Binary Optimization (QUBO) problems. Specifically, we investigate the application of Quantum Computing and Machine Learning to improve the performance of optimization algorithms. Our study employs Matrix Sparsification, Graph Spanner, and Graph Neural Networks to streamline the complexity of QUBO formulations. Through extensive experimentation, we demonstrate that these approaches significantly reduce computational costs and improve solution quality when implemented on quantum computing platforms like the D-Wave Leap Hybrid Solver. Integrating machine learning techniques, particularly in graph neural networks, further refine problem-solving, enabling near-optimal solutions within constrained time limits. This interdisciplinary research advances the field of quantum-inspired computing and holds substantial potential for practical applications in business optimization and logistics.

Keywords: Quantum Computing / QUBO (Quadratic Unconstrained Binary Optimization) / Matrix Sparsification / Optimization Algorithms / Quantum-Inspired Computing / Computational Efficiency / Machine Learning / TSP (Travelling Salesperson Problem)

หัวข้อปริญญานิพนธ์	การเพิ่มประสิทธิภาพเทคนิคการคำนวณที่ได้รับแรงบันดาลใจจากควอนตัม สำหรับการแก้ปัญหาประเภท Quadratic Unconstrained Binary Optimization
หน่วยกิต	3
ผู้เขียน	นายปณิธิ สุวรรณโณ นายมุฮัมมัด บินฮาร์
อาจารย์ที่ปรึกษา	ผศ.ราชวิชช์ สโรชวิกสิต
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2566

บทคัดย่อ

งานวิจัยนี้ใช้เทคนิคการคำนวณที่ได้รับแรงบันดาลใจจากควอนตัมเพื่อเพิ่มประสิทธิภาพการคำนวณในการแก้ปัญหา Quadratic Unconstrained Binary Optimization (QUBO) เราได้ศึกษาการประยุกต์ใช้การคำนวณเชิงควอนตัมและการเรียนรู้ของเครื่องเพื่อปรับปรุงประสิทธิภาพของอัลกอริทึมการเพิ่มประสิทธิภาพ การศึกษาของเราใช้การทำ Matrix Sparsification, Graph Spanner และ Graph Neural Networks เพื่อลดความซับซ้อนของปัญหาโจทย์ Traveling Salesperson Problem (TSP) ที่ถูกแปลงให้ในรูปแบบ QUBO จากการทดลองของเรา เราแสดงให้เห็นว่าแนวทางเหล่านี้ช่วยลดต้นทุนการคำนวณและปรับปรุงคุณภาพของคำตอบเมื่อดำเนินการบนแพลตฟอร์มการคำนวณเชิงควอนตัมเช่น D-Wave Leap Hybrid Solver นอกเหนือจากนี้การรวมเทคนิคการเรียนรู้ของเครื่องโดยเฉพาะอย่างยิ่งใน Graph Neural Networks ยังช่วยเพิ่มความสามารถในการแก้ปัญหาให้ได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดภายในเวลาที่จำกัด งานวิจัยนี้ไม่เพียงแต่พัฒนาสาขาการคำนวณเชิงที่ได้รับแรงบันดาลใจจากควอนตัมแต่ยังมีศักยภาพอย่างมากในการประยุกต์ใช้จริงในด้านการเพิ่มประสิทธิภาพทางธุรกิจและโลจิสติกส์ต่อไปในอนาคต

คำสำคัญ: คอมพิวเตอร์ควอนตัม / QUBO / การแยกส่วนเมทริกซ์ / อัลกอริธึมการปรับให้เหมาะสม / คอมพิวเตอร์ที่ได้รับแรงบันดาลใจจากควอนตัม / ประสิทธิภาพการคำนวณ / การเรียนรู้ของเครื่อง / TSP

ACKNOWLEDGMENTS

We extend our heartfelt gratitude to Asst. Prof. Rajchawit Sarochawikasit for his invaluable guidance, extensive expertise, and dedicated mentorship throughout the course of this project. His advanced knowledge in High-Performance Computing and Quantum Computing, particularly with the D-Wave platform, has been a cornerstone in shaping the trajectory of our research. We appreciate Asst. Prof. Sarochawikasit's commitment to sharing his wealth of knowledge, even as he navigates the complexities of these cutting-edge technologies. His unwavering quantum research aspiration and modest action have been instrumental in shaping our understanding and approach, not only enriching our experience but also setting a benchmark for continued contributions to the advancements in quantum computational techniques.

Our sincere thanks also go to Quantum Technology Foundation Thailand (QTFT) for their unwavering support, providing resources, and fostering an environment conducive to groundbreaking research in quantum technologies. Additionally, we express our gratitude to Vorapong Suppakitpaisarn for his exceptional mentorship at QTFT. His guidance has been instrumental in navigating the intricacies of quantum-inspired algorithms, contributing significantly to the success of this project. Including Sippanat Achavananthadidth and Theeramet Sinthumongkolchai, who have provided mathematical support as well as attention since the first day of work-integrated learning. This collaborative effort with Asst. Prof. Rajchawit Sarochawikasit and QTFT team have been a valuable and enriching experience, and we look forward to continued contributions to the advancements in quantum-inspired computational techniques.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$G(V, E)$	Graph
$\mathcal{Q}$	QUBO matrix
$\mathcal{L}$	Loss Function

LIST OF TECHNICAL VOCABULARY AND ABBREVATIONS

QUBO	=	Quadratic Unconstrained Binary Optimization
TSP	=	Travelling Salesperson Problem
TTS	=	Time-to-solution
ML	=	Machine Learning
GNN	=	Graph Neural Network
GCN	=	Graph Convolutional Network
BQM	=	Binary Quadratic Model
CQM	=	Constrained Quadratic Model
DQM	=	Discrete Quadratic Model

CHAPTER 1 INTRODUCTION

1.1 Problem Statement, Motivation, and Potential Benefits

Across a multitude of disciplines such as logistics, finance, and resource allocation, the optimization of complex combinatorial problems remains an essential endeavor. Traditional classical optimization approaches often encounter limitations in addressing intricate interdependencies and inherent constraints inherent in these challenges. Bridging this gap necessitates the effective utilization of Quadratic Unconstrained Binary Optimization (QUBO) problems. Central to this discussion is the seamless integration of QUBO formulations with established mathematical tools and advanced high-performance computing methodologies, enhancing their capacity to conquer intricate optimization hurdles.

The motivation for embarking on this research journey derives from the pressing need to unlock the potential of Quantum Unconstrained Binary Optimization (QUBO) in tackling complex optimization conundrums. Quantum-inspired optimization introduces a compelling paradigm that seamlessly aligns with the principles of quantum mechanics. QUBO problems involve assigning binary values to variables, with the goal of minimizing a quadratic function, a framework that closely resonates with the foundational concepts of quantum computing. This motivation is grounded in the understanding that QUBO problems hold the promise of revolutionizing optimization paradigms by amalgamating quantum principles with sophisticated mathematical techniques and strategies in high-performance computing. This interdisciplinary fusion amplifies the efficiency of QUBO algorithms and translates quantum insights into practical solutions for real-world challenges.

One specific avenue of exploration within this broader context is the utilization of Matrix Sparsification, an approach that aims to reduce the complexity of QUBO matrices. Through the convergence of quantum-inspired linear algebra and high-performance computing, our aim is to unlock optimization solutions that transcend classical methodologies. This Quantum Advantage materializes as quantum principles facilitate revolutionary optimization transformations, parallel exploration of solutions, and expedited convergence. By infusing quantum insights and leveraging matrix sparsification techniques, we aspire to redefine the thresholds of optimization excellence, thereby addressing intricate challenges across a spectrum of domains.

The potential dividends of this research endeavor are multifaceted. Primarily, by integrating Matrix Sparsification into QUBO formulations, we anticipate elevating the quality and computational efficiency of optimization solutions, potentially surpassing traditional methods. Moreover, this research fosters the synergy between quantum-inspired optimization and real-world problems, driving industrial innovation and advancing quantum computing and optimization theory. The outcomes of this project hold the potential to find practical applications in diverse domains, ranging from optimizing supply chains to enhancing the efficiency of energy distribution. In essence, this research contributes not only to scholarly knowledge but also fuels industrial innovation by harnessing the transformative power of quantum optimization and matrix sparsification for real-world problem-solving.

1.2 Project Type

Theoretical research-based project. Our project is fundamentally characterized as a theoretical research-based endeavor. It is anchored in the pursuit of new knowledge, insights, and innovative solutions within the realm of Quadratic Unconstrained Binary Optimization (QUBO) and quantum-inspired techniques.

In essence, our project type embodies the spirit of intellectual exploration, where theoretical rigor, innovative thinking, and the quest for deeper understanding converge to advance the frontiers of knowledge and contribute to the academic and scientific community's collective wisdom.

1.3 Proposed Method

1.3.1 Approach

Our approach is rooted in the integration of quantum-inspired techniques to enhance Quadratic Unconstrained Binary Optimization (QUBO) solutions. We recognize the transformative potential of quantum principles and aim to apply them effectively to QUBO challenges. This involves the development of algorithmic strategies that draw inspiration from quantum mechanics, optimizing QUBO formulations to achieve a Quantum Advantage. By refining convergence, enhancing solution quality, and broadening problem-solving capabilities, our approach seeks to redefine QUBO optimization and provide more efficient and precise solutions to complex real-world challenges.

1. Quantum-Inspired Techniques: Our project is initiated by delving into quantum-inspired techniques tailored to the QUBO framework. These techniques are designed to enrich the QUBO formulation and improve its efficiency in solving complex optimization problems.

2. Algorithm Development and Implementation: At the core of our approach lies the development and rigorous implementation of pioneering algorithms, leveraging both the Qiskit quantum computing development platform and the D-Wave quantum annealer. These algorithms are meticulously designed to harness quantum-inspired insights and principles to optimize Quadratic Unconstrained Binary Optimization (QUBO) formulations. Within the Qiskit environment, we engineer and implement algorithms tailored to the intricacies of QUBO problems. Capitalizing on Qiskit's extensive quantum computing toolkit, these algorithms are designed to expedite convergence and enhance solution quality. The utilization of quantum-inspired techniques within Qiskit facilitates the exploration of complex optimization challenges, aligning our work with quantum computing principles. In parallel, we extend our algorithmic solutions to the D-Wave quantum annealer, a specialized quantum computing platform optimized for optimization tasks. This dual-pronged approach allows us to assess the performance and efficiency of our algorithms across distinct quantum computing environments. By adapting our algorithms for execution on D-Wave, we aim to explore their versatility and robustness, providing a comprehensive evaluation of their applicability to different quantum platforms. This integrated section emphasizes our commitment to algorithm development and implementation, highlighting our work's adaptability to both Qiskit and D-Wave and its alignment with quantum computing principles across diverse quantum platforms.

3. Quantitative Performance Evaluation: An integral facet of our approach entails a comprehensive quantitative assessment of the developed algorithms. This performance evaluation encompasses crucial metrics, including solution quality, computational efficiency, and convergence speed.

4. Real-world Applications and Case Studies: Our proposed approach will be applied to real-world optimization challenges spanning diverse domains such as logistics, finance, and resource allocation. Through a series of case studies, we aim to demonstrate the practical effectiveness of our quantum-inspired approach in solving intricate optimization problems characterized by multiple variables and constraints.

1.3.2 Objectives

1. To Achieve Profound Understanding (Comprehensive Exploration of QUBO Matrix Optimization Principles): Gain a profound understanding of Quantum Unconstrained Binary Optimization (QUBO) principles and their adaptability to solve intricate optimization challenges across various domains.

2. To Enhance Effectiveness (Integration of Quantum Techniques): Integrate quantum-inspired techniques into the QUBO optimization framework to refine problem formulations and enhance their effectiveness in addressing complex real-world optimization problems.

3. To Optimize Algorithm Performance (Efficient Algorithm Development): Develop highly efficient QUBO algorithms that leverage integrated mathematical approaches and high-performance computing strategies for optimal problem encoding and accelerated solution exploration.

4. To Apply in Practical Context (Application to Real-World Optimization): Apply the developed QUBO algorithms to a diverse range of real-world optimization challenges, assessing how the integration of quantum-inspired techniques and computational strategies enhances solution quality and efficiency across different industries.

5. To Quantify and Compare (Quantitative Performance Assessment and Comparative Analysis): Conduct a comprehensive quantitative evaluation of the QUBO algorithms, considering key performance metrics such as solution quality, convergence speed, and computational efficiency. Additionally, implement quantum annealing on platforms like D-Wave and perform a rigorous comparative analysis against classical solvers, taking into account integrated mathematical and computational approaches to determine the advantages of our methodology.

1.3.3 Scope

1. Research and Understanding

   - In-Depth Study of QUBO Principles - A thorough examination of Quantum Unconstrained Binary Optimization (QUBO) principles and their foundational theories, aimed at gaining a profound understanding of their theoretical underpinnings.
   - Exploration of Mathematical Foundations - An investigation into the mathematical foundations of linear algebra, encompassing diverse techniques and their applicability to optimization, to establish a solid knowledge base.

2. Integration of Techniques

   - Incorporation of Quantum-Inspired Techniques - Integration of advanced quantum-inspired techniques into the QUBO optimization framework with the objective of enhancing problem formulations and optimizing the solution process.
   - Exploration of Matrix Sparsification - Investigating matrix sparsification techniques as a means to reduce problem complexity and enhance the efficiency of QUBO optimization.
   - Possibility of AI-Assisted Sparsification - We are actively exploring the potential of incorporating artificial intelligence (AI) techniques for matrix sparsification. AI methods, such as machine learning algorithms, may provide innovative approaches to identify and retain crucial elements within the QUBO matrices, further enhancing the effectiveness of sparsification strategies.

3. Algorithm Development

- Quantum-Inspired Algorithm Design - We embark on the creation of QUBO algorithms inspired by principles from the realm of quantum computing. These algorithms are tailored to leverage quantum-inspired techniques to optimize problem formulations, thereby enhancing solution quality and convergence.
- Matrix Sparsification Integration - Matrix sparsification techniques are systematically integrated into our algorithms, enabling the reduction of problem complexity by emphasizing essential problem components. This integration aims to streamline the optimization process and improve algorithm efficiency.
- Efficient Encoding Strategies - Our approach prioritizes the development of encoding strategies that efficiently map intricate optimization problems onto binary variables. These encoding techniques are designed to ensure optimal problem representation, facilitating enhanced algorithm performance.

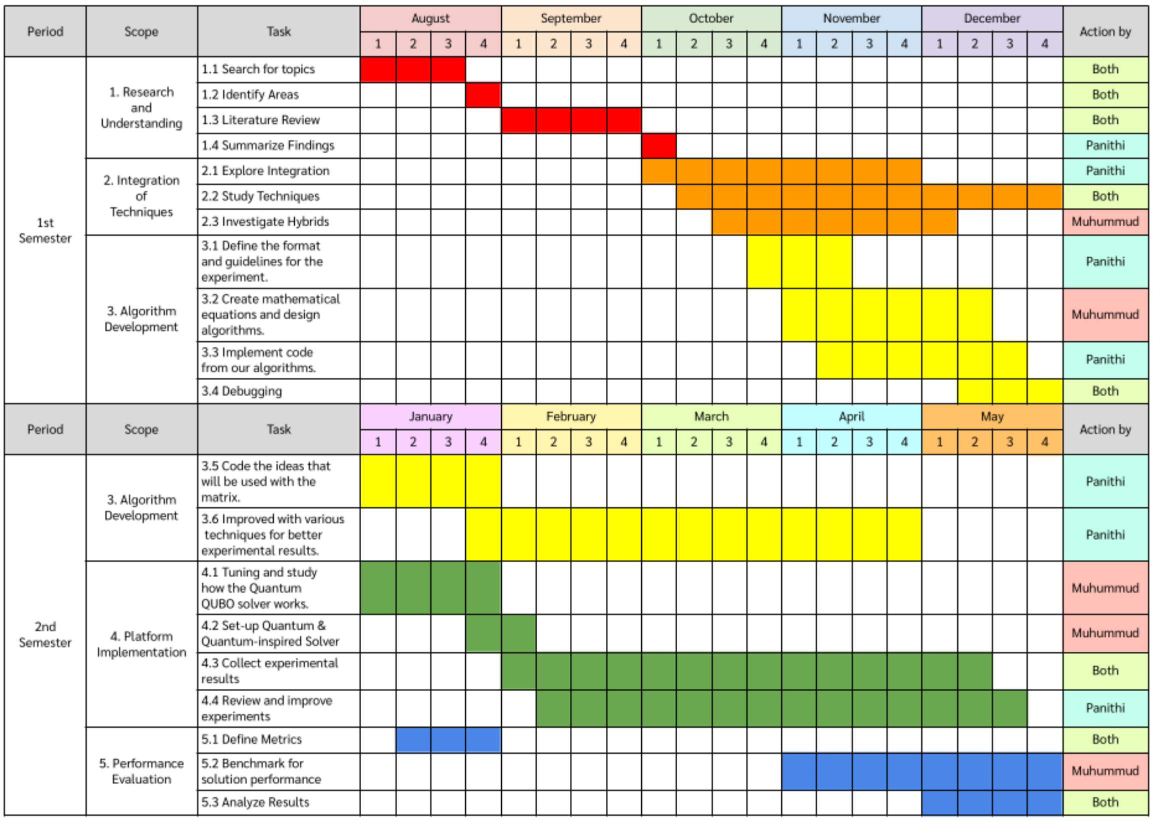
4. Platform Implementation Our platform implementation strategy encompasses mathematical and AI methods, emphasizing robust quantum annealing on various platforms, with a particular focus on leveraging D-Wave for quantum computing exploration. The implementation plan unfolds in the following key dimensions:

   - Quantum Annealing on Diverse Platforms - We extend our research by implementing quantum annealing techniques on various quantum computing platforms, notably D-Wave. This strategy enables a comprehensive evaluation of algorithm performance in quantum environments and facilitates a comparative analysis against classical solvers.
   - Cross-Platform Analysis - Rigorous examination of how our integrated techniques impact quantum and classical optimization approaches. This analysis elucidates the advantages and limitations of our methodology, providing insights into the adaptability of our algorithms across different computational paradigms.
   - Optimizing Quantum Resources - Directing our efforts towards optimizing the utilization of quantum computing resources, with a specific emphasis on leveraging D-Wave. This strategic approach ensures that our implementation aligns seamlessly with our research goals, maximizing the efficiency of quantum-inspired transformations.
   - Real-world Application Simulation - Simulating real-world application scenarios like Travelling Salesperson Problem to evaluate the practical efficacy of our quantum annealing methodologies. This involves testing our algorithms on problem instances inspired by actual use cases, providing a tangible assessment of their performance and applicability in practical settings.

5. Performance Evaluation

   - Quantitative Assessment - Rigorous quantitative evaluation of algorithm performance, taking into account critical factors such as convergence speed and solution accuracy, to gauge the efficacy of our methodology.
   - Comparative Analysis - A comparative analysis of results obtained from QUBO optimization using different techniques and solvers, shedding light on the impact of integrated methodologies on the optimization outcomes.
   - Scalability Analysis - Explore the scalability of the proposed methodology by assessing its performance across a range of problem sizes. Investigate how well the algorithm handles increasingly larger instances of the QUBO for Traveling Salesperson Problem.

1.3.4 Schedule



Period	Scope	Task	August				September				October				November				December				Action by
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1st Semester	1. Research and Understanding	1.1 Search for topics	🟥	🟥	🟥																		Both
		1.2 Identify Areas				🟥																	Both
		1.3 Literature Review					🟥	🟥	🟥	🟥													Both
		1.4 Summarize Findings									🟥												Panithi
	2. Integration of Techniques	2.1 Explore Integration									🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧					Panithi
		2.2 Study Techniques										🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	Both
		2.3 Investigate Hybrids											🟧	🟧	🟧	🟧	🟧	🟧	🟧				Muhummud
	3. Algorithm Development	3.1 Define the format and guidelines for the experiment.												🟨	🟨	🟨							Panithi
		3.2 Create mathematical equations and design algorithms.													🟨	🟨	🟨	🟨	🟨	🟨			Muhummud
		3.3 Implement code from our algorithms.														🟨	🟨	🟨	🟨	🟨	🟨		Panithi
		3.4 Debugging																		🟨	🟨	🟨	Both

Period	Scope	Task	January				February				March				April				May				Action by
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
2nd Semester	3. Algorithm Development	3.5 Code the ideas that will be used with the matrix.	🟨	🟨	🟨	🟨																	Panithi
		3.6 Improved with various techniques for better experimental results.				🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨					Panithi
	4. Platform Implementation	4.1 Tuning and study how the Quantum QUBO solver works.	🟩	🟩	🟩	🟩																	Muhummud
		4.2 Set-up Quantum & Quantum-inspired Solver				🟩	🟩																Muhummud
		4.3 Collect experimental results					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Both
		4.4 Review and improve experiments						🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		Panithi
	5. Performance Evaluation	5.1 Define Metrics		🟦	🟦	🟦																	Both
		5.2 Benchmark for solution performance													🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	Muhummud
		5.3 Analyze Results																	🟦	🟦	🟦	🟦	Both

Figure 1.1 Work plan for both semesters of the Senior Project.

CHAPTER 2 BACKGROUND THEORY AND LITERATURE REVIEWS

In this pivotal chapter, we embark on a scholarly journey through the intricate landscape of background knowledge and an expansive exploration of the existing literature. Our aim is to cultivate a profound understanding of the foundations that underpin our research endeavor. The discourse will traverse various domains, elucidating key concepts, theoretical frameworks, and foundational principles that contribute to the intellectual landscape of our project. The background knowledge section will meticulously unfold layers of fundamental concepts relevant to Quantum Unconstrained Binary Optimization (QUBO) problems, delving into the theoretical underpinnings of quantum computing and its practical implications. This segment serves as a conceptual scaffold upon which the subsequent chapters will build, ensuring a robust foundation for our research. Simultaneously, our literature review endeavors to map the intellectual terrain traversed by previous researchers and scholars in the realm of quantum-inspired optimization, matrix sparsification, and artificial intelligence. Each review is a stepping stone, revealing the evolution of thought, methodologies employed, and key findings in the pertinent domains. This meticulous examination of existing scholarship not only informs our research trajectory but also positions our work within the broader academic discourse.

Furthermore, this chapter goes beyond a mere enumeration of facts and studies; it offers critical insights, identifies gaps in the existing body of knowledge, and delineates the rationale for our chosen research direction. By critically evaluating prior work, we lay the groundwork for the innovative contributions our project aspires to make, establishing its significance within the larger academic and practical context. In essence, this chapter serves as the compass guiding our intellectual exploration, ensuring that the subsequent chapters are not only well-informed but also poised to push the boundaries of knowledge in the realms of quantum-inspired optimization, matrix sparsification, and artificial intelligence.

2.1 Overview

The Quadratic Unconstrained Binary Optimization (QUBO) problem, characterized by its applicability across diverse domains from finance to machine learning, stands as a formidable challenge within the realm of combinatorial optimization. Its NP-hard nature poses a computational hurdle for classical problem-solving methodologies. However, a transformative avenue emerges through the lens of quantum computing, where QUBO finds a solution path via the method of quantum annealing within the framework of adiabatic quantum computation. This quantum-inspired approach opens new horizons for tackling the complexity of QUBO, providing a promising trajectory for overcoming classical computational limitations and unlocking optimized solutions in various practical applications. QUBO is essential in providing the ability to formulate many combinatorial problems in the form of:

$$\text{Minimize } y = x^T Q x; x \in \{0,1\}^n$$

Where x is the binary decision and Q is an n-by-n square, symmetric of real-valued coefficients. The general form of QUBO can be expressed as:

$$\text{Minimize } y = \sum_i a_i x_i + \sum_{i,j} b_{i,j} x_i x_j$$

- x_i represent the binary variable with the value of 0 or 1
- a_i is the linear bias associated with variable x_i
- $b_{i,j}$ denotes the coefficient of the quadratic interaction between variables x_i and x_j

2.2 Literature Review

2.2.1 Quantum Bridge Analytics I: Formulating and Using QUBO Models

The paper explores the field of Quantum Bridge Analytics (QBA) [3], which is dedicated to developing tools and methods that facilitate the synergy between classical and quantum computing. This integration is envisioned to unlock the combined potential of both computing paradigms for immediate and future practical applications. Structured as a two-part tutorial series, this paper constitutes the first part. Its primary objective is to comprehensively survey the key components of Quantum Bridge Analytics and their practical applications. Notably, it places a strong emphasis on enhancing understanding through numerical illustrations. Within the tutorial's framework, Part 1 focuses its attention on the Quadratic Unconstrained Binary Optimization (QUBO) model. This model holds paramount significance in the realm of quantum computing, as it serves as the prevailing optimization model. Remarkably, it unifies a diverse spectrum of combinatorial optimization problems, offering a versatile foundation for quantum computational endeavors.

Combinatorial Optimization (CO) is a pivotal domain within the broader field of optimization, holding significant relevance across various industries, encompassing both private enterprises and governmental sectors. This field stands as a focal point for active research conducted by communities in Operations Research, Computer Science, and Analytics. These research endeavors aim to devise and evaluate novel techniques for tackling practical CO problems that require decision-making under conditions where numerous yes/no choices result in associated objective function values, such as costs or profits. Addressing these problems poses considerable challenges. Historically, the conventional approach to CO problems involved crafting specialized solution algorithms tailored to the specific mathematical structures of individual problems. While this approach has yielded favorable outcomes in certain problem domains, it has its limitations, chiefly the requirement to develop a multitude of distinct solution techniques, each with a narrow scope of applicability. In recent years, there has been a remarkable discovery regarding the Quadratic Unconstrained Binary Optimization (QUBO) formulation. QUBO, as an acronym for this formulation, has been shown to encompass a wide array of significant CO problems encountered in industry, science, and government sectors, as substantiated by studies like Kochenberger et al. (2014) and Anthony et al. (2017). Importantly, through readily applicable reformulation techniques, the efficacy of QUBO solvers can be harnessed to efficiently address numerous pivotal problems once they are cast within the QUBO framework.

The QUBO model has emerged as a foundational element within quantum computing, particularly in quantum annealing and Fujitsu's digital annealing approaches. It has also garnered attention in the realm of neuromorphic computing. These connections have positioned QUBO models at the core of experimentation conducted with quantum computers developed by D-Wave Systems, as well as neuromorphic computers advanced by IBM. This intersection between QUBO models and quantum computing has spurred exploration efforts by prominent organizations, encompassing IBM, Google, Amazon, Microsoft, D-Wave, and Lockheed Martin in the commercial domain, and institutions like Los Alamos National Laboratory, Oak Ridge National Laboratory, Lawrence Livermore National Laboratory, and NASA's Ames Research Center in the public sector. Collectively, the computational experiences amassed by both classical and quantum computing communities underscore not only the potential but also the effectiveness of the QUBO model as an alternative to conventional modeling and solution methodologies, thus opening new avenues for addressing complex optimization challenges.

The paper discusses the connection between Quantum Bridge Analytics (QBA) and the Quadratic Unconstrained Binary Optimization (QUBO) model. QBA seeks to bridge classical and quantum computing, and the QUBO model plays a key role in this effort. The QUBO model is a mathematical framework that can represent a wide range of combinatorial optimization problems, making it a versatile tool for both classical and quantum computing. It has gained importance in quantum annealing and neuromorphic computing, forming the basis for experiments with quantum and neuromorphic computers developed by various organizations. The paper illustrates how optimization problems from various domains can be reformulated as QUBO models. This includes showing how constraints can be integrated into the QUBO framework using penalty functions and providing precise model representations. The paper also explores recent innovations in solving QUBO models, which have applications in classical and quantum computing, as well as machine learning.

In summary, the paper highlights the utility of the QUBO model in bridging classical and quantum computing through Quantum Bridge Analytics, demonstrating its versatility in solving a wide array of optimization problems and its potential impact on various fields.

2.2.2 Quantum Bridge Analytics II: Network Optimization and Combinatorial Chaining for Asset Exchange

Quantum Bridge Analytics (QBA) focuses on developing tools to bridge classlical and quantum computing for immediate gains and future applications in quantum computing. In Part I of this tutorial, the Quadratic Unconstrained Binary Optimization (QUBO) model was highlighted as a crucial component, unifying various combinatorial optimization problems and becoming the dominant optimization model in quantum computing.

Part II of the tutorial [4], presented in this paper, introduces an application that enhances the utility of QUBO models. It does so by coordinating QUBO solutions through the Asset Exchange Problem (AEP), which encompasses numerous important optimization applications. The AEP allows individuals or institutions to identify and execute mutually beneficial asset exchanges, akin to positive-sum games in game theory. This approach leverages the power of QUBO models, using their solutions as inputs or outputs to solve a broader spectrum of problems. Motivated by the Quantum Bridge Analytics perspective, this paper provides formulations and methodologies for the AEP, collectively referred to as Asset Exchange Technology. This technology offers tools to facilitate positive-sum games, similar to mechanisms such as money, interest, and middlemen. Notably, Asset Exchange Technology aligns with blockchain goals, aiming to reduce reliance on intermediaries. Asset Exchange Technology holds significance within the Quantum Bridge Analytics context, particularly considering that quantum computing is expected to remain in its early stages for the next decade. Developing commercial applications for near-term quantum computing through hybrid classical-quantum techniques, a focus of QBA is crucial. Innovations in these hybrid techniques provide fertile ground for introducing Asset Exchange Technology. Additionally, the paper draws connections between Asset Exchange Technology and portfolio optimization, highlighting how QUBO models form a basis for integrating classical and quantum computing through the AEP. The AEP extends beyond portfolio optimization, fostering cooperative optimization networks among holders of multiple portfolios. This naturally aligns with QUBO models, which identify desirable assets for different participants. Asset Exchange Technology, utilizing network optimization and combinatorial chaining, finds exchanges that meet diverse desirability criteria to benefit all participants. The components of network optimization and combinatorial chaining within Asset Exchange Technology are integrated within the netform modeling framework. Netform modeling characterizes structural aspects of models to enhance insight and solution effectiveness, as observed in previous works. The paper proceeds by presenting examples of asset exchange applications, followed by the formulation of the basic AEP problem and its transformation into a network optimization model. It then explores the structure of combinatorial chaining in relation to this network model and introduces advanced processes that merge network optimization, combinatorial chaining, and metaheuristic analysis for addressing complex asset exchange instances.

In conclusion, Quantum Bridge Analytics (QBA) has demonstrated its practical relevance by addressing the challenging Asset Exchange Problem (AEP). This perspective opens up a multitude of applications in fields such as financial investment, resource allocation, economic distribution, and collaborative decision-making. The combination of network optimization and metaheuristic optimization through combinatorial chaining leads to the development of an Asset Exchange Technology capable of tackling a wide range of practical variations. Although current quantum computers have limitations, particularly in handling small AEP problems due to limited qubits, the integration of network formulations and combinatorial chaining paves the way for addressing significantly larger AEP problems. The AEP model represents a crucial class of optimization problems that can be effectively approached within the QBA domain. This provides a foundational framework for future advancements, particularly as quantum computing technology matures.

2.2.3 Quadratic Unconstrained Binary Optimization Problem Preprocessing: Theory and Empirical Analysis

This paper discusses the Quadratic Unconstrained Binary Optimization problem (QUBO), which serves as a versatile model for various combinatorial optimization problems and fosters interdisciplinary connections. It highlights a new type of quantum annealing computer that maps QUBO onto a physical qubit network with specific size and edge density constraints, gaining increasing attention for reducing the underlying QUBO structure. The paper presents rules for diminishing the size of the QUBO matrix by identifying variables whose optimal values can be predetermined. These reductions result in improved solution quality and reduced time to find solutions. Moreover, for metaheuristic methods, which may not guarantee optimal solutions, these reductions enhance the quality of solutions obtained within reasonable time limits. In this research, the main motivation lies in the versatility of the Quadratic Unconstrained Binary Optimization problem (QUBO) framework, which is used to model a wide range of optimization problems. The recent prominence of QUBO in the context of modern quantum annealing computers has spurred the need for more efficient preprocessing methods to reduce the size of the Q matrix and thus speed up the process of finding good solutions.

QUBO problems are known to be NP-complete, which means they are computationally challenging. However, modern metaheuristics have proven effective in finding good solutions to large QUBO problems. Additionally, a new type of quantum computer, based on quantum annealing and employing a physical network structure of qubits called a Chimera graph as shown in Figure 2.1, has demonstrated the ability to rapidly find good solutions to QUBO problems. The Chimera structure in quantum computing consists of a connected network of qubits, where groups of qubits are densely interconnected. Still, these groups are sparsely connected to other groups of densely connected qubits. This structure can be likened to social network visualizations or a collection of densely connected cities connected to other cities through fiber optic backbones. This unique architecture has shown promise in solving QUBO problems efficiently.

The research focuses on business problems, often involving big data, which are characterized by their unstructured nature and randomness. To better capture these types of problems, the authors have created new test problems that contain sparsely connected elements, with most elements uniformly distributed but with varying amounts of outlier elements. The primary contribution of this research is the formulation and validation of five rules for reducing the size of QUBO. These rules have been combined into a fast preprocessing tool called QPro, which has been tested with both the exact solver Cplex and a tabu search metaheuristic incorporating path relinking. The results of this testing demonstrate that the key factors influencing reduction are the range of uniformly distributed elements and the number and magnitude of quadratic outliers.

In conclusion, the QPro preprocessing tool is shown to be highly efficient and effective in reducing the time required to obtain high-quality solutions. The research also highlights the potential for applying these rules in sensitivity analysis to achieve robustness and identify transformations that can increase the size of the problem while reducing edge density per node.

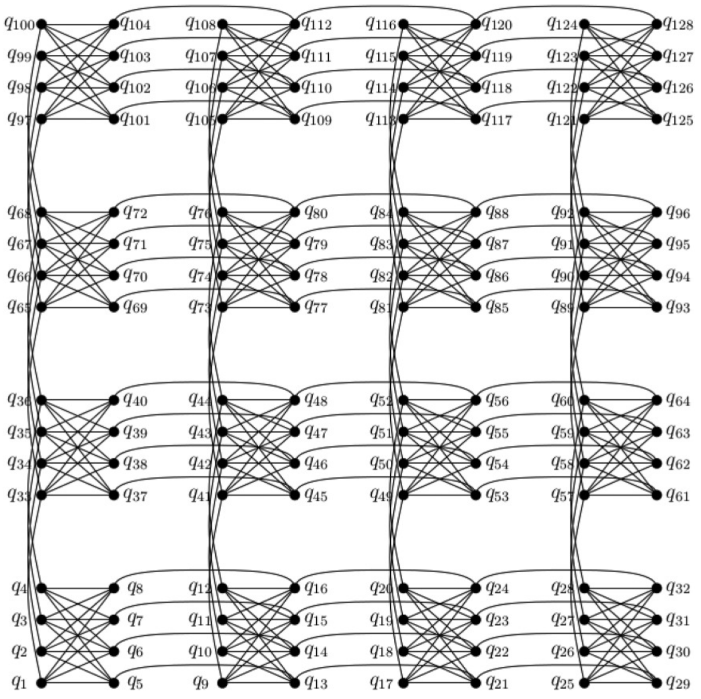


Figure 2.1 Example Chimera Network Structure [1].

2.2.4 Solving workflow scheduling problems with QUBO modeling

The introduction of the paper discusses the growing interest in quantum computing, both in industry and academia, with the goal of creating specialized hardware capable of simulating quantum processes that are classically intractable. It outlines two primary paradigms for constructing algorithms using quantum bits (qubits): the gate model and the adiabatic quantum computing model. These models manipulate qubits to perform logical operations and evolve a system to represent a final function, respectively. These models are shown to be polynomially equivalent and have been extensively studied. And advancements in quantum hardware development have led to prototypes accessible via companies like Google, IBM, and D-Wave Systems. These companies offer cloud-based access to quantum processing units (QPUs) tailored for various quantum algorithms, aimed at tackling complex problems in fields such as machine learning, simulation, and optimization. Notably, optimization has seen significant progress with quantum annealing and quantum approximate optimization algorithms (QAOA), both implementable on current quantum hardware.

This paper highlights a gap in understanding regarding how quantum algorithms can impact combinatorial optimization without error correction. While error correction could enable faster quantum algorithms, it's unclear if noisy intermediate-scale quantum (NISQ) computing can overcome limitations. Hybrid quantum-classical algorithms have arisen as an interim solution. Variational algorithms have been used in quantum machine learning, quantum chemistry, and optimization tasks. The paper's focus is on comparing hybrid algorithms and other techniques to address scheduling problems, specifically the workflow scheduling problem. The structure of the paper is outlined, with sections dedicated to explaining scheduling problems and previous quantum computing research, formally introducing the investigated workflow scheduling version and its modeling as a QUBO for quantum optimization, presenting experimental data and algorithms used, discussing results, and concluding the study.

One well-known scheduling problem is the job-shop scheduling problem (JSP), where N jobs need to be executed on M machines with no additional constraints, and it is already NP-hard. This paper highlights the use of an Ising model for studying JSP in the context of quantum computing. Another example discussed is the Nurse Scheduling problem, which deals with scheduling nurses to shifts considering their availability and other constraints. An illustrative example of a scheduling problem with 6 nodes and limited resources at each time slot is provided in Figure 2.2. The goal of solving this problem is to generate a makespan map, which depicts when each job is completed. Figure 1c displays a sub-optimal makespan map for this particular problem, while Figure 1d shows an optimal makespan map. The comparison between these two maps demonstrates the difference between sub-optimal and optimal solutions in terms of job completion times.

In the conclusion of the paper, the authors presented a novel approach to tackling a specific type of scheduling problem called the workflow scheduling problem by formulating it as a Quadratic Unconstrained Binary Optimization (QUBO) problem. They drew inspiration from a real-world scenario and expanded on existing implementations of similar scheduling problems to incorporate more realistic constraints. The focus was on addressing cases where job dependencies and maximum resource capacities needed to be considered. The introduction of these constraints significantly increased the complexity of the QUBOs. To handle this, the authors explored decomposition techniques to solve these QUBOs using various quantum and hybrid solvers. Through their investigation, they observed that hybrid and classical algorithms were more successful in solving the instances, while quantum solvers struggled even with the smallest problems.

The authors highlighted the impact of the decomposition technique, emphasizing that the challenges in solving scheduling and optimization problems using quantum methods are intricate and extend beyond the inherent complexity of the problems themselves. They noted that by reducing the problem size, some of these limitations could be mitigated. Future efforts were identified in two main directions. Firstly, the authors proposed exploring specific sub-classes of scheduling problems that could be more efficiently represented in QUBO form. Additionally, they expressed interest in developing novel implementations of hybrid quantum and quantum-inspired algorithms, aiming to better address the QUBOs that arise from real-world instances of scheduling problems.

2.2.5 Combinatorial Optimization on Gate Model Quantum Computers

In this paper, the author mainly discusses the possibility of exploiting and solving QUBO on Gate Model Quantum Computers. Firstly, QUBO solution can be executed on an adiabatic quantum computer (AQC) with a physical process called quantum annealing e.g. D-Wave quantum computers. This paper provided ways to adapt the Qubit gate to solve the QUBO problem [5]:

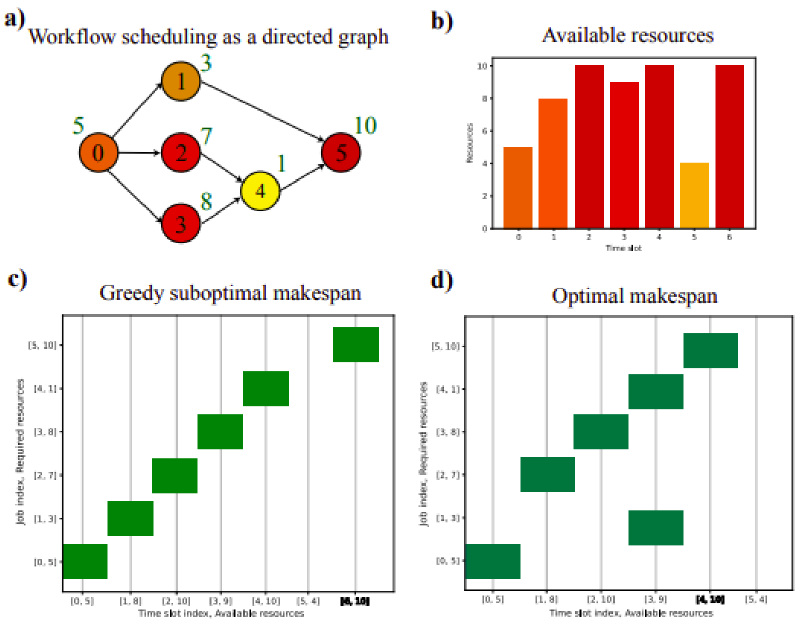


Figure 2.2 Example of a workflow scheduling problem with 6 jobs [2].

1. Simulating AQC on a gate model quantum computer - The reason that adiabatic quantum computing should be simulated into a gate model computer is due to the limitation of architecture that currently cannot accommodate any type of non-stochastic Hamiltonian. However, a problem can be decomposed into the terms in a generic Hamiltonian into a sequence of single- and two-qubit gates. The idea of simulating AQC is to decompose the unitary evolution of a quantum system into single-qubit and two-qubit gates.

2. Global quantum optimization

   - Grover-based quantum global optimization Grover's search algorithm has shown superior performance with a quadratic speed-up over the best classical algorithm. This leads to the idea that Grover's algorithm would provide BQP (bounded error quantum polynomial time) to NP problems. To apply Grover's algorithm to the problem, oracles are used to find the best solution by comparing new and old solutions and then producing the best solution as output.

   - A quantum search heuristic - By adopting the idea of Hamiltonian that changes its state from initial state to final state, the quantum heuristic search algorithm uses the sequences of unitary operators as a representation of quantum system evolution:

$$M = U^{(j)}P^{(j)} \dots U^{(1)}P^{(1)}$$

   - Where the operator M acts on the initial state, at the k-th step, the operator U^k and P^k is the mixing and phrase operators.

3. Quantum algorithms to approximate the ground state of the Ising model - This section illustrates the set of quantum algorithms that provide the approximation of the ground state. This set of algorithms is based on the introduction of some unknown parameters into the unitary evolution of the system and the subsequent finding of the optimal value of these parameters so as to maximize the overlap between the resultant quantum ground state under the evolution and the true ground state of the system

4. Quantum metropolis sampling - This method is sampling from low energy levels of the quantum system which are usually used to find the backbone of the combinatorial problem and enable the capability to reduce the original optimization problem into smaller subproblems. Further application can be found in estimating the ground state of an Ising model Hamiltonian.

This study gave examples of ways to demonstrate that addressing combinatorial optimization problems on a gate model quantum computer is doable by utilizing a unified programming framework (such as QUBO). Their goal is that by combining concepts like these, researchers will be able to develop more advanced algorithms that can harness the power of gate model quantum computers to address difficult combinatorial optimization issues.

2.2.6 Logical and inequality implications for reducing the size and di□culty of quadratic unconstrained binary optimization problems

The paper focuses on addressing the Quadratic Unconstrained Binary Optimization (QUBO) problem, which arises in various optimization contexts, including Ising spin problems, graph theory, and binary discrete optimization. The paper emphasizes the importance of preprocessing techniques to transform the QUBO problem's graph representation into a smaller, equivalent graph. This reduction in graph size is valuable for enhancing the quality and speed of solutions obtained using both exact and metaheuristic algorithms. Additionally, it is a crucial step towards adapting large-scale QUBO problems for execution on quantum annealing computers with specific hardware graphs [6].

Their research findings underscore the significant utility of the preprocessing rules, particularly when applied to QUBO problems characterized by structures and densities commonly encountered in intricate real-world applications. It is noteworthy that this approach, while highly effective, stands out as a noteworthy alternative to the highly refined preprocessing method employed by CPLEX, yielding a substantial reduction in the number of variable settings required.

Across an extensive testbed comprising 96 distinct problem instances, the rules consistently demonstrated their effectiveness by efficiently assigning many variables a priori. This resulted in the generation of substantially smaller problem instances. In approximately half of the problems within the test bed, the approach, QPRO+, achieved a remarkable 45 percent reduction in problem size and impressively solved 10 problems exactly. Furthermore, the rules excelled in identifying numerous implied relationships between pairs of variables, leading to the formulation of numerous straightforward logical inequalities.

The computational testing, conducted using the algorithm outlined in this paper and an experimentally designed test bed of 96 problems, revealed several noteworthy observations:

1. Problems characterized by sparser structures exhibited a greater propensity for reduction compared to denser problems.

2. Larger problem instances demanded more processing time to achieve reductions.

3. Surprisingly, a smaller number of linear outliers in the problem instances resulted in more substantial reductions, while variations in the magnitude and quantity of quadratic outliers had minimal impact.

4. Rule 1.0: If $c_i + D_i^- \geq 0$ or > 0, then $x_i = 1$ is uniquely optimal, and Rule 3.4: Assume $d_{i,h} \geq 0$. If $-c_i - c_h - d_{i,h} - D_i^- - D_h^- \leq 0$ then $x_i = x_h \geq 1$ and moreover $x_i = x_h = 1$ in an optimal QUBO solution. Both collectively contributed to 25 percent of the reductions acheived.

5. The majority of reductions occurred during the initial processing passes.

2.2.7 Graph Analysis: Graph Sparsification

The paper discusses the challenge of processing large graphs efficiently and explores the concept of graph sparsification as a potential solution. In traditional thinking, it was assumed that as the number of nodes in a graph grows, the number of edges would also grow linearly. Specifically, it was believed that the number of edges |E| would be less than or equal to some constant multiple of the number of vertices $|V|$, denoted as $|E| \leq c|V|$. However, in practical scenarios, it has been observed that the relationship between the number of edges and vertices is more complex. In practice, it often follows a polynomial growth pattern $|E| = |V|^{1+c}$, with a small constant c, such as $c = 0.1$ (although sometimes as high as $c = 0.5$). This means that the number of edges grows faster than linearly in relation to the number of vertices.

The central question addressed in the paper is whether it is possible to reduce the graph to a subset of edges denoted E' in such a way that the number of edges $|E|$ becomes proportional to the logarithm of the number of vertices, $|V| \log |V|$. In essence, the goal is to represent the graph in a more compact form, taking up significantly less space while still allowing for similar analytical capabilities. This reduction to a sparser representation, if achievable, would have practical implications for the efficient processing of large graphs, making complex analyses more manageable.

Graph sparsification, as discussed in the paper, holds significant relevance within the context of Quantum Unconstrained Binary Optimization (QUBO) projects and related research. The core idea revolves around efficiently handling large graphs, a common challenge encountered in both classical and quantum computing environments:

1. Data Representation Enhancement - In the realm of QUBO projects and the preceding QUBO-related paper, the presence of extensive datasets or intricate optimization problems represented as graphs is commonplace. The management of such sizable graphs can be arduous. Graph sparsification techniques offer a practical solution by enabling the representation of these graphs in a more concise form. This reduction in edge density is achieved while retaining essential structural characteristics, thus optimizing the use of computational resources.

2. Optimizing Quantum Computing Resources - Quantum computing is not impervious to resource limitations, including the availability of qubits and quantum gates. The practicality of processing large graphs directly on quantum hardware may be constrained. By applying graph sparsification strategies, the size of the graph can be effectively reduced while preserving pertinent information. This approach enhances the feasibility of leveraging quantum resources for addressing optimization problems associated with QUBO formulations.

3. Simplified Problem Complexity - QUBO instances, as they grow in size, can become computationally intensive, leading to challenges in finding timely solutions. The application of graph sparsification simplifies the problem by diminishing the dimensions and intricacies of the underlying graph structure. Consequently, QUBO instances can be addressed with greater efficiency on both classical and quantum computing platforms. This efficiency is critical for achieving high-quality solutions within reasonable timeframes.

4. Quantum Computing Integration - Quantum computers are well-suited for tackling specific types of optimization problems, including those cast as QUBOs. Graph sparsification contributes to the alignment

of QUBO projects with quantum computing by reducing the intricacy of the optimization problem. This compatibility enhancement is particularly valuable for quantum annealers, such as those provided by D-Wave Systems, where sparser graph representations can lead to expedited and more precise problem-solving processes.

In a scholarly context [7], graph sparsification emerges as a valuable technique for managing the computational challenges presented by expansive graphs, especially in the domain of optimization problems like QUBOs. Its capacity to enhance resource utilization, diminish computational intricacies, and facilitate the fusion of quantum computing methodologies into QUBO projects underscores its strategic significance. This approach synergizes effectively with the objectives and objectives of both your QUBO project and the preceding QUBO-oriented research paper, promising advancements in the field of combinatorial optimization.

CHAPTER 3 RESEARCH METHODOLOGY

In this pivotal chapter, we traverse the intricate landscape of Methodology, sculpting a well-defined path for our research odyssey aimed at enhancing Quantum-Inspired Computational Techniques for Quadratic Unconstrained Binary Optimization (QUBO) problem-solving. This comprehensive exploration seamlessly merges theoretical frameworks with practical implementations, ensuring a robust approach to contribute substantively to the evolving domain of quantum computing research.

Before delving into the intricacies of our two-phased methodology, it is crucial to articulate the foundational hypothesis guiding our research. At the heart of our endeavor lies the hypothesis that the strategic utilization of QUBO matrix sparsification, geared towards reducing its complexity, holds the promise of substantial enhancements in the computation and processing time when implemented on quantum-inspired computers.

The motivation behind this hypothesis stems from the inherent challenges posed by complex QUBO problems, especially when executed on quantum-inspired platforms. As quantum computing, with its promise of parallelism and exponential speedup, continues to evolve, the need to streamline problem formulations becomes paramount. Our hypothesis postulates that by employing matrix sparsification strategies, we can effectively distill the essential elements of QUBO problems, facilitating a more efficient exploration of solution spaces on quantum-inspired devices.

The focus of our attention on matrix sparsification is driven by the understanding that a sparse matrix, containing predominantly zero elements, allows for a more targeted and streamlined optimization process. The hypothesis further contends that through the judicious reduction of non-essential matrix components, the quantum-inspired algorithms can navigate solution spaces more swiftly, resulting in tangible reductions in both computation time and processing resources.

In the pursuit of validating this hypothesis, our methodology unfolds in two distinct phases. The initial pre-computational phase involves the meticulous exploration of matrix sparsification strategies, with the Traveling Salesperson Problem (TSP) serving as a benchmark testbed. This carefully chosen problem archetype enables a comprehensive evaluation of our hypothesis, leveraging well-established datasets and optimal solutions to benchmark the effectiveness of various sparsification techniques.

The subsequent computational phase translates our findings into tangible implementations on quantum-inspired devices, notably D-Wave's platform. By validating our hypothesis through real-world quantum-inspired computation, we aim not only to enhance the understanding of matrix sparsification but also to contribute valuable insights into optimizing quantum-inspired algorithms for practical problem-solving scenarios [8]. As we traverse through these two phases, the outcomes promise not only advancements in quantum-inspired computational techniques but also a deeper comprehension of the transformative potential that matrix sparsification holds for the realm of optimization.

3.1 Pre-Computational Phase: Matrix Sparsification Strategies

In the intricate landscape of matrix sparsification for enhancing computational techniques, our pre-computational phase embarks on a journey marked by diversity and innovation. As we delve into strategies to refine the representation of the Traveling Salesperson Problem (TSP), we are presented with a multifaceted array of techniques, each offering a unique perspective on how to distill the essential elements of optimization challenges.

The decision to explore not one, but three distinct avenues of matrix sparsification arises from a commitment to comprehensiveness and a recognition of the nuanced nature of optimization problems. In our quest for efficiency, we navigate through these methodologies with a shared goal – to streamline the representation of TSP instances while retaining the critical information necessary for meaningful optimization.

As we unfold this chapter, you will witness the convergence of classical methods, advanced artificial intelligence, and innovative graph-theoretic approaches. Each pathway holds the promise of reshaping the computational landscape, and together, they contribute to a holistic understanding of matrix sparsification's transformative potential.

Join us on this exploration as we unravel the rationale behind each approach, the unique advantages they bring, and the synergies that emerge when diverse techniques harmonize. By doing so, we aim not only to refine our understanding of matrix sparsification but also to equip ourselves with a versatile toolkit for optimizing a spectrum of real-world challenges. Our journey begins with a thoughtful consideration of three distinct sparsification strategies, each poised to leave an indelible mark on the landscape of optimization.

3.1.1 Matrix Sparsification by Disregarding Terms

3.1.1.1 Technique Overview

This approach involves a strategic reduction of computational complexity by disregarding certain terms within the distance matrix representation of the Traveling Salesperson Problem (TSP). The core principle lies in establishing a threshold value, beyond which terms in the distance matrix are considered non-essential and consequently rounded to zero. This results in the creation of a sparse matrix, where numerous elements become negligible, facilitating a more efficient computation process.

3.1.1.2 Rational

- Efficiency Gains - The primary objective of this technique is to expedite the optimization process by simplifying the distance matrix. By designating a threshold value, terms that contribute minimally to the overall solution are effectively pruned, reducing the matrix's computational load.

- Simplicity and Speed - One of the inherent strengths of this method lies in its simplicity and rapid application. The straightforward process of rounding terms below the threshold to zero ensures a quick pre-processing step, making it an attractive option for scenarios where computational efficiency is paramount.

3.1.1.3 Methodology

- Threshold Definition - A critical aspect of this technique is the careful selection of the threshold value. This value is often determined through empirical testing or by considering the characteristics of the TSP instance. Fine-tuning the threshold allows for a balance between sparsity and retention of critical information.

- Implementation - Once the threshold is defined, the implementation involves traversing the distance matrix and rounding terms below the threshold to zero. This step-by-step process transforms the original dense matrix into a sparse representation.

- Adaptability - The adaptability of this technique is noteworthy. Its simplicity makes it applicable to a wide range of optimization challenges beyond TSP, provided the problem can be expressed as a distance matrix.

3.1.1.4 Evaluation and Validation

- Quantitative Metrics - The effectiveness of this technique is evaluated through quantitative metrics such as computation time, solution quality, and the degree of matrix sparsity. A comparative analysis against baseline methods and other sparsification techniques provides insights into its relative performance.

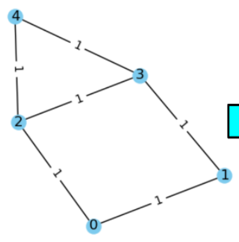
- Sensitivity Analysis - Given its reliance on a threshold value, a sensitivity analysis is conducted to explore how variations in this parameter impact the efficiency gains. This analysis helps fine-tune the method for optimal performance across diverse TSP instances.

3.1.1.5 Expected Outcomes and Contributions

- Sparse Matrix Generation - Successful implementation of this technique results in a sparse representation of the distance matrix, preserving essential information while expediting subsequent optimization algorithms.

- Computational Efficiency - Anticipated outcomes include notable gains in computational efficiency, making it an attractive preprocessing step for TSP instances and potentially extending its applicability to other optimization challenges.

- Versatility - The simplicity and adaptability of this technique position it as a versatile tool in the matrix sparsification toolkit, offering insights into the trade-off between matrix density and computational speed.



$$\Rightarrow \begin{bmatrix} 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3.1 The nature of the matrix is initially very dense, then we make it more sparse by setting a threshold value.

3.1.2 Graph Spanner for Matrix Sparsification

3.1.2.1 Technique Overview

The application of a Path-Greedy t-Spanner involves the strategic construction of a subgraph that optimally preserves distances between cities [9] in the Traveling Salesperson Problem (TSP). This technique employs a path-greedy approach to create a t-Spanner, a subgraph that not only ensures essential connections between cities but also facilitates more efficient exploration of solution spaces during TSP computations.

3.1.2.2 Rational

- Structured Exploration - Unlike traditional approaches, the Path-Greedy t-Spanner technique strategically alters the TSP graph's structure. Let $G(V, E)$ be the original graph representing cities and distances, and $G_t(V_t, E_t)$ be the t-Spanner subgraph, where E_t is a subset of E. The path-greedy strategy ensures that connections in G_t are established in a manner that optimally influences subsequent TSP optimization processes.

- Path-Greedy Strategy - Denoting the distance between two cities i and j as $d_{i,j}$, the path-greedy strategy involves iteratively adding edges to G_t. At each iteration, the edge (i, j) is added if $d_{i,j}$ is among the t smallest distance form i to j. This adaptability makes the technique suitable for a diverse range of TSP instances.

3.1.2.3 Methodology

- t-Spanner Construction
  - The t-Spanner construction process involves iteratively selecting edges based on the path-greedy strategy. Let $d_{i,j}$ represent the distance between cities i and j. At each iteration, the algorithm identifies the t smallest distances from i to j and adds the corresponding edge to G_t.

$$E_t = \{(i, j)|d_{i,j} \text{ is among the } t \text{ smallest distance from } i \text{ to } j\}$$

  - This process continues until G_t satisfies the t-Spanner properties, ensuring that the ratio of distances in G_t is at most t times greater than the original graph G.
- Adaptation to TSP
  - Integrating the Path-Greedy t-Spanner into TSP instances involves utilizing G_t as the guiding subgraph during the optimization process. The structured connections in G_t significantly reduce unnecessary computations, providing a more efficient exploration of solution spaces.
- Parameter Tuning
  - Fine-tuning parameters, especially the stretch factor (t), allows for adaptation to different characteristics of TSP instances. The stretch factor governs the trade-off between the sparsity of G_t and the preservation of essential connections.

3.1.2.4 Evaluation and Validation

- Comparison with Baseline - To evaluate the performance of the Path-Greedy t-Spanner technique, we compare it against baseline TSP computations. Quantitative metrics, such as computation time (τ) and solution quality (Q), provide insights into the efficiency gains:

$$\text{Efficiency Gain} = \frac{\text{Computation time of baseline} - \text{Computation time of t-Spanner}}{\text{Computation time of baseline}}$$

- Versatility Assessment - The adaptability of this technique is assessed by applying it to TSP instances with varying complexities (C). A diverse set of scenarios ensures a comprehensive evaluation of its versatility and efficiency gains.

3.1.2.5 Expected Outcomes and Contributions

- Guided Exploration - Successful implementation of the Path-Greedy t-Spanner technique results in a TSP graph (G_t) that guides subsequent computations towards a more efficient exploration of solution spaces.

- Computational Efficiency - Anticipated outcomes include reduced computation time (τ_t) and enhanced solution quality (Q_t), , showcasing the technique's potential to contribute to the optimization of TSP instances.

- Applicability Beyond TSP - The adaptability of the Path-Greedy t-Spanner technique positions it not only as a valuable tool for TSP optimization but also as a potential strategy for other graph-based optimization challenges.

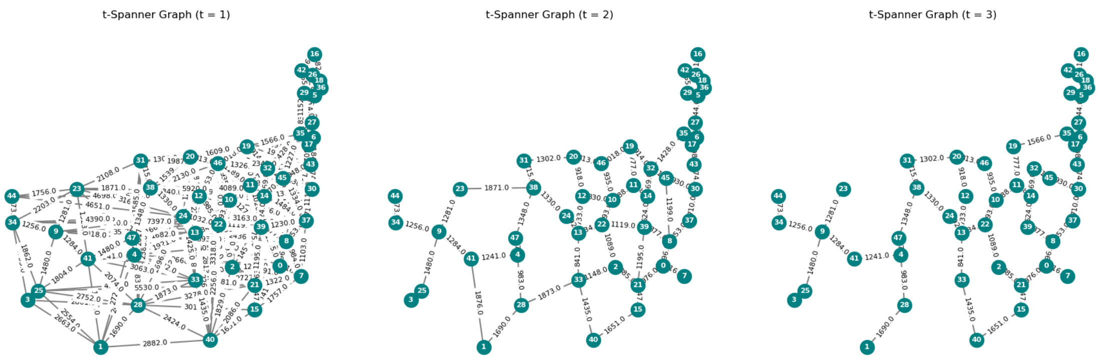


Figure 3.2 Testing our t-spanner with data from TSPLIB95 named ATT48.

3.1.3 Graph Neural Networks (GNNs)

The TSP is a classic combinatorial optimization problem known for its NP-hard nature, requiring the exploration of a vast solution space to find the optimal route. Traditional ML approaches may struggle with the combinatorial explosion of possibilities. And TSP inherently involves a graph structure where cities are nodes and distances are edges. Conventional ML models might overlook the complex dependencies and spatial relationships encoded in this graph structure. So, ML models without a graph-based understanding may not effectively capture the spatial context crucial in TSP [10][11]. GNNs, designed for graph-structured data, excel in retaining and utilizing spatial information.

3.1.3.1 Technique Overview

Graph Neural Networks (GNNs) are a sophisticated class of neural networks tailored to analyze and extract meaningful patterns from graph-structured data [12]. In the context of the Traveling Salesperson Problem (TSP) with QUBO formulation, the integration of GNNs aims to dynamically learn and encode the intricate relationships and dependencies within the TSP graph [13], ultimately contributing to the generation of an optimized QUBO matrix.

3.1.3.2 Rational

- The rationale behind incorporating GNNs lies in their inherent capacity to capture complex spatial dependencies and patterns within graph-structured data [14].

- By employing GNNs for QUBO matrix generation, the method adapts to the specific structural nuances of TSP instances, potentially outperforming traditional, handcrafted matrix sparsification approaches.

3.1.3.3 Methodology

1. Graph Representation - Represent the TSP graph as an undirected graph $G(V, E)$, where V is the set of cities (nodes) and E is the set of edges representing distances between cities.

2. Node Features - Encode node features, X_i, for each city i, , potentially incorporating spatial coordinates and other relevant characteristics.

3. Edge Features - Encode edge features, $E_{i,j}$, to capture the distance information between cities i and j.

4. GNN Architecture Design - Design a GNN architecture [15] with graph convolutional layers (GCN) and attention mechanisms, aiming to capture spatial dependencies and relationships within the TSP

graph:

$$H^{l+1} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

- $\mathbf{H}^{(l+1)}$: Node representation at layer $l+1$.
- σ: Activation function (commonly ReLU or another non-linear function).
- $\hat{\mathbf{D}}^{-\frac{1}{2}}$: Degree matrix $\hat{\mathbf{D}}$ raised to the power of -0.5.
- $\hat{\mathbf{A}}$: Adjacency matrix with added self-connections.
- $\mathbf{H}^{(l)}$: Node representation at layer l.
- $\mathbf{W}^{(l)}$: Weight matrix at layer l.

5. Training Data Preparation - Utilize the dataset obtained from previous experiments, forming input-output pairs for GNN training: (X, E, Q), where X is the node feature matrix, E is the edge feature matrix, and Q is the target QUBO matrix.

6. Supervised Learning - Train the GNN in a supervised learning setting, optimizing the QUBO matrix prediction by minimizing the mean squared error:

$$\mathcal{L}(Q_{pref}, Q) = \frac{1}{2N^2}\sum_{i,j}(Q_{pred}^{i,j} - Q^{i,j})^2$$

In the context of machine learning and regression problems, particularly when dealing with the QUBO matrix prediction task for the Traveling Salesperson Problem (TSP), Q_{pred} stands for the predicted QUBO matrix.

- Q_{pred} is the predicted QUBO matrix obtained from the Graph Neural Network (GNN) or any other predictive model.
- Q is the target or ground truth QUBO matrix.

The loss function $\mathcal{L}$ quantifies the difference between the predicted QUBO matrix and the actual QUBO matrix, and the goal of the training process is to minimize this loss. Minimizing the MSE encourages the model to generate QUBO matrices that closely match the ground truth, improving the accuracy of the model's predictions.

3.1.3.4 Evaluation and Validation

- Quantitative Metrice - Evaluate the performance of the trained GNN using metrics such as mean squared error (MSE) and accuracy.
  - Time-to-Solution - Measure the time required for the QUBO formulation to converge to a solution. This metric provides insights into the computational efficiency of the proposed method.
  - Solution Quality - Evaluate the quality of the obtained solutions by comparing them to known optimal or benchmark solutions. This ensures that, despite the speed improvements, the proposed method does not compromise on solution quality.
- Comparison with Baseline - Compare the generated QUBO matrices with baseline matrices obtained from traditional handcrafted methods.
  - Traditional Methods - Compare the results obtained using the GNN-based QUBO formulation with baseline results obtained from traditional handcrafted methods. Highlight any improvements in terms of both time-to-solution and solution quality.

  - Quantitative Analysis - Conduct a quantitative analysis, presenting statistical measures such as mean, median, and standard deviation for both time-to-solution and solution quality. This provides a comprehensive overview of the method's performance.

3.1.3.5 Expected Outcomes and Contributions

- Improved Matrix Generation - The GNN is expected to generate QUBO matrices (Q_{pred}) that accelerate the time-to-solution for TSP instances, indicating an enhancement in the optimization process.
  - Faster Convergence - Anticipate and showcase that the GNN-based QUBO formulation leads to faster convergence to feasible solutions for TSP instances compared to traditional methods.
  - Trade-off Considerations - Discuss any trade-offs made in terms of solution optimality to achieve faster convergence. Clearly articulate the balance between computational efficiency and solution quality.
- Data-Driven Insights - The GNN may reveal intrinsic structural insights within TSP instances, providing valuable knowledge for future optimization strategies and problem understanding.
  - Graph Understanding - Emphasize the potential for GNNs to provide data-driven insights into the underlying structure of TSP instances. Explore how learned representations might reveal patterns and relationships within the TSP graph.
  - Generalization - Discuss the generalizability of insights gained from GNNs to different TSP instances. Explore whether the learned features capture universal characteristics of TSP instances.
- Enhanced Time-to-Solution - Anticipated outcomes include a reduction in the time required to obtain feasible TSP solutions using the QUBO formulation, showcasing the potential of GNNs in optimizing combinatorial problems.
  - Real-World Implications - Discuss the real-world implications of achieving enhanced time-to-solution. Explore how faster computations can positively impact decision-making processes in scenarios where TSP optimization is critical.
  - Scalability - Address the scalability of the proposed approach. Discuss how the method performs as the size and complexity of TSP instances increase.

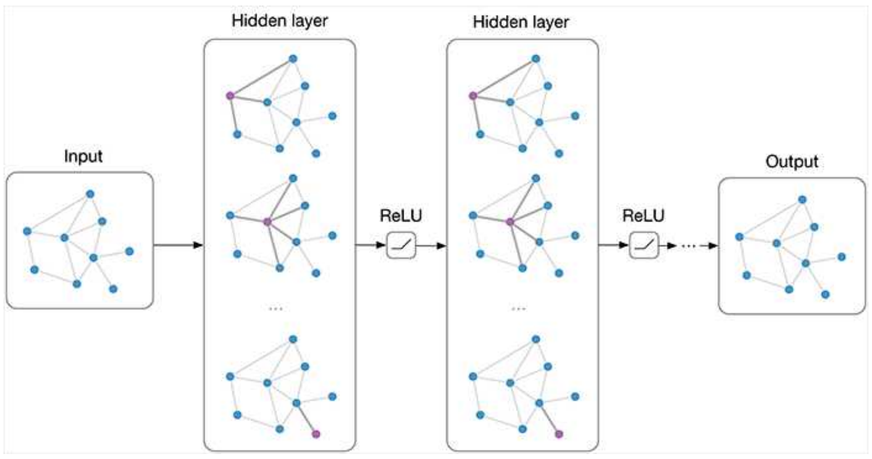


Figure 3.3 Example of GNN with ReLU non-linear activated function.

3.2 Computational Phase: Implementation with D-Wave, Bridging Theory and Practice

The prepared QUBO-matrix from the Pre-computational process will be mapped to suit 3 different machines; D'wave 2000Q, D'wave Hybrid Solver BQM, and CQM. All of the experiments will be run with the same annealing time of 20 µs about 30 times or more, thus they can be collected as statistical data.

Size	Dataset	$\|V\|$
Large	RD100	100
Medium	ATT48	48
Small	GR17	17

Table 3.1 The dataset selected from the TSPLIB95 for our experiment is as follows.

3.2.1 D'Wave 2000Q

Some parameters need to be adjusted before performing the calculations on QPU such as chain strength and annealing time. However, D'wave has provided a function "scaled" that can adjust chain strength to the number of qubits used and default annealing time of 20 µs and running with 1000 samples.

Pseudo-code for quantum annealing:

Algorithm 1 Solving a BQM using DWaveSampler

```
1: Initialize sampler with DWaveSampler and use EmbeddingComposite
2: Sample QUBO using sampler with specified parameters
3: Show sampleset using DWave inspector
4: Extract problem ID and chain strength from sampleset
5: Write problem ID to output file
6: Iterate over sampleset data sorted by energy
7: Extract sample, energy, and other parameters from each sample
8: Build solution from sample
9: Check if solution is valid
10: Compute score of solution
11: Write results to solution file
12: Break loop if solution found
13: Calculate chain break fraction if no solution found
14: Write appropriate message to output file
```

3.2.2 D'Wave BQM and CQM

The QUBO-matrix can be mapped through BQM by using D'wave provided functions `BinaryQuadraticModel()` and `BinaryQuadraticModel.from_qubo()`.

For CQM, the function `ConstrainedQuadraticModel.from_quadratic_model()` will be used to transform the binary quadratic model into a form of CQM objective function.

After performing the experiment on all 3 machines, results will be collected to illustrate the performance by comparing our processed QUBO matrix to our original TSP matrix, focusing on the quality of the answer given the same annealing time and the same quality of answer regardless of annealing time. If the processed QUBO-matrix yields a good enough solution with a faster annealing time, our hypothesis will be accepted.

Pseudo-code for implementing QUBO-matrix on BQM:

Algorithm 2 Solving a BQM using Leap Hybrid Sampler

```
Require: QUBO matrix
Ensure: solution array
       Initialize $bqm$ with the provided QUBO matrix
    2: Sample $bqm$ using Leap hybrid sampler with a time limit, store result in $sampleset$
       Print elapsed time
    4: Retrieve and print best sample and energy from $sampleset$
       Map best sample to $solution$ array
```

CHAPTER 4 EXPERIMENT RESULTS

The culmination of our research project is revealed in Chapter 4, where we set out on a journey through the maze of experiments and painstakingly write a story that sheds light on the important lessons learned from the furnace of hard testing. We march relentlessly through the maze of optimization approaches, applying a variety of classical and quantum-inspired techniques with the dexterity of seasoned craftspeople.

Our journey begins with a symphony of algorithms, each chord resonating with the promise of discovery and innovation. Through the lens of experimentation, we orchestrate a multifaceted tableau, a tapestry interwoven with threads of classical wisdom and quantum intrigue. At the heart of this symphony lies the pursuit of truth, the quest to validate hypotheses born from the crucible of intellectual inquiry.

As we delve deeper into the labyrinth, we encounter the first glimmer of enlightenment: matrix sparsification by naively disregarding terms, a bold gambit that echoes the audacity of pioneers charting uncharted territories. With the precision of master artisans, we chip away at the edifice of complexity, sculpting a landscape of clarity and efficiency. In this realm of exploration, every discarded term is a testament to the indomitable spirit of inquiry, a beacon guiding us towards the elusive shores of optimization excellence.

Yet, our odyssey does not end here, for we are adventurers bound by an insatiable thirst for knowledge. Venturing further into the abyss, we uncover the enigmatic realm of graph theory, where paths intertwine and possibilities abound. Here, we invoke the arcane arts of path-greedy graph spanners, wielding their power to forge a new paradigm of edge reduction. Through the alchemy of computation, we distill complexity into simplicity, unraveling the intricate tapestry of connectivity with each calculated step.

But our journey is far from over, for on the horizon lies the realm of artificial intelligence, a realm teeming with untapped potential and boundless creativity. Here, we harness the power of machine learning to unveil the secrets of sparsification patterns hidden within the annals of data. Guided by the wisdom of algorithms and the intuition of human intellect, we unlock the vaults of knowledge, revealing insights that transcend the boundaries of conventional wisdom.

In Chapter 4, we invite you to join us on this epic voyage of discovery, where each experiment is a testament to the indomitable human spirit and the relentless pursuit of truth. Together, let us chart a course towards a future illuminated by the brilliance of innovation and the transformative power of knowledge.

4.1 Pre-Computational: Algorithms for QUBO Sparsification

4.1.1 Hypothesis Testing with Greedy Disregarding Terms

Our experimentation begins with the implementation of traditional algorithms, such as greedy algorithms, to establish a baseline for comparison. These algorithms, while straightforward, provide valuable insights into the performance of conventional optimization strategies in addressing Quadratic Unconstrained Binary Optimization (QUBO) problems.

Subsequently, we explore the impact of matrix sparsification techniques on the optimization process. By naively disregarding certain terms within the QUBO matrix, we aim to systematically reduce its complexity and enhance the efficiency of subsequent optimization algorithms. This process enables us to test our hypothesis that enhancing matrix sparsity will expedite the time-to-solution of quantum-inspired solvers, such as the D'Wave quantum annealer.

Furthermore, to validate our findings and ensure the robustness of our approach, we leverage advanced optimization solvers such as Gurobi. Operating on local computing resources, Gurobi offers a powerful platform for testing and refining optimization algorithms before deployment on quantum devices. Given the limited access and complexities associated with quantum hardware, this preliminary testing phase allows us to iterate rapidly and fine-tune our methodologies without constraints.

By meticulously documenting our experimental procedures and meticulously analyzing the results, we aim to provide valuable insights into the efficacy of different optimization techniques for solving complex combinatorial problems. Through this comprehensive approach, we strive to push the boundaries of optimization research and pave the way for the practical application of quantum-inspired computational techniques in real-world scenarios.

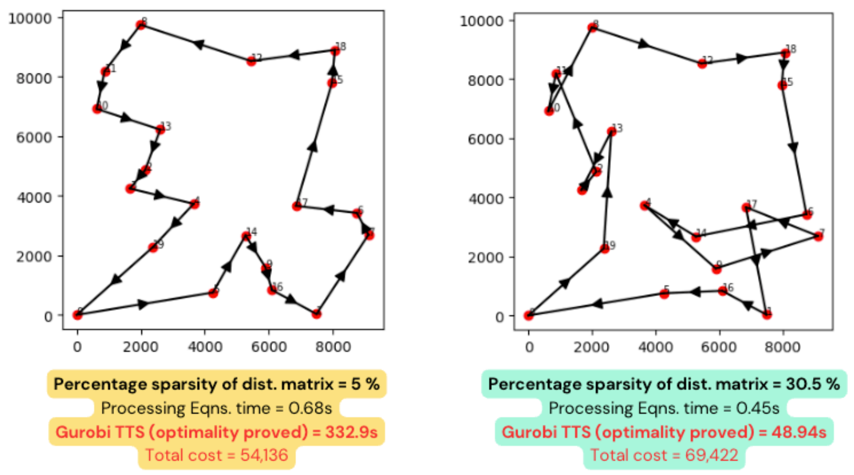


Figure 4.1 Testing our assumption on matrix sparsification can make QUBO calculation faster.

In this set of experiments conducted on Gurobi as show in Figure 4.1, a notable observation emerges from the comparison of results. Random sampling was employed to generate a set of 20 sample points, thereby inducing increased sparsity within the distance matrix. Upon juxtaposing the experimental outcomes, it becomes evident that heightened matrix sparsity correlates with expedited runtime performance. However, a compelling aspect arises as the quality of obtained solutions exhibits a notable reduction. Consequently, this observation serves as a corroborative testament to our experimental hypothesis, positing that augmenting matrix sparsity yields the potential to mitigate the prolonged runtime predicament encountered in Quadratic Unconstrained Binary Optimization (QUBO) instances.

Moreover, an intriguing inquiry arises regarding the methodology employed to induce matrix sparsity. While the present experiment utilizes a simplistic greedy approach to selectively disregard lower-order terms within the matrix, the quest for identifying optimal sparsification techniques warrants further investigation. This necessitates a deeper exploration into diverse sparsification strategies, as the current approach merely scratches the surface of potential methodologies. Thus, future research endeavors will be dedicated to scrutinizing alternative sparsification methodologies, with the aim of optimizing the trade-off between computational efficiency and solution quality in QUBO problem-solving scenarios.

4.1.2 Harness Graph Theory with Path-greedy Spanner

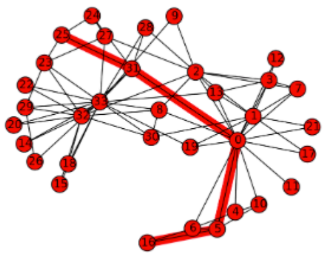
Within combinatorial optimization, the effectiveness of different approaches is always being investigated to address the intrinsic complexity of Quadratic Unconstrained Binary Optimization (QUBO) issues. Graph theory is one such promising field where the use of graph spanners seems as a convincing method to improve problem solving abilities. Graph spanners provide a new way to improve matrix sparsity in the QUBO framework. They are well-known for their capacity to maintain important graph features at a substantial edge count reduction.

The concept of graph spanners revolves around the idea of constructing a subgraph that retains crucial connectivity properties of the original graph while minimizing the number of edges. By strategically selecting a subset of edges that efficiently capture the essential structural characteristics of the graph, graph spanners present a potent tool for sparsifying QUBO matrices. This reduction in edge density not only facilitates expedited computational processing but also holds the potential to enhance solution quality by focusing computational resources on salient graph features.

Moreover, the utilization of graph spanners in the context of QUBO optimization introduces a fascinating interplay between graph theory and combinatorial optimization. By leveraging insights from graph spanner theory, researchers can devise innovative strategies to transform intricate QUBO instances into more manageable problem formulations. This interdisciplinary fusion not only enriches the repertoire of optimization techniques but also opens new avenues for synergistic exploration between disparate fields.

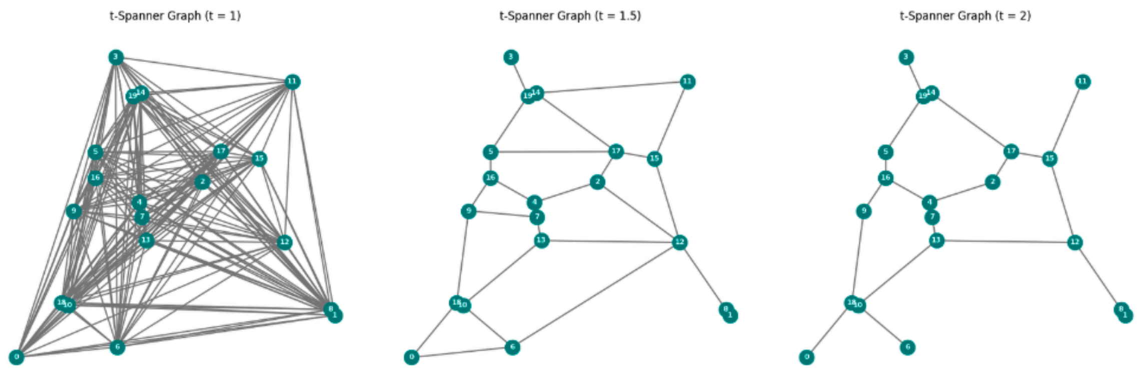
Let $G = (V, E)$ be the original, and $G' = (V, E')$ be the t-spanner subgraph.

For all $u, v \in V$, the shortest path length between u and v in G' never exceeds t times the shortest path length between u and v in G.



$$\forall\ u, v \in V,\ \text{if}\ \delta_G(u,v)\ \text{is the shortest path length between}\ u\ \text{and}\ v\ \text{in}\ G,\ \text{then}\ \delta_{G'}(u,v) \leq t{\times}\delta_G(u,v)$$

In the upcoming section, we embark on an exploration of graph spanners within the QUBO optimization paradigm on Figure 4.2. Through rigorous experimentation and analysis, we aim to elucidate the efficacy of graph spanners in enhancing computational efficiency and solution quality in QUBO problem-solving scenarios. By delving into the intricacies of graph spanner theory and its application in the context of QUBO optimization, we endeavor to uncover novel insights and methodologies to propel the frontier of combinatorial optimization forward.



$$\forall\ u, v \in V,\ \text{if}\ \delta_G(u,v)\ \text{is the shortest path length between}\ u\ \text{and}\ v\ \text{in}\ G,\ \text{then}\ \delta_{G'}(u,v) \leq t{\times}\delta_G(u,v)$$

Figure 4.2 Sparsification can be viewed as sparse with more principles than previous method.

Building upon the foundation laid by previous methodologies, the integration of graph spanners into the QUBO optimization framework represents a natural evolution towards achieving greater matrix sparsity.

While the previous approach of greedily disregarding terms provided a rudimentary means of reducing matrix complexity, the utilization of graph spanners offers a more nuanced and mathematically rigorous approach to sparsification. By selectively retaining edges that preserve essential graph properties, graph spanners ensure that the resulting subgraph maintains structural integrity while significantly reducing edge density. This strategic sparsification process, rooted in mathematical principles, not only accelerates computational processing but also maintains solution optimality by focusing computational resources on critical graph features. Thus, the incorporation of graph spanners into the QUBO optimization paradigm builds upon the principles of previous methods while advancing the field towards more sophisticated and mathematically grounded approaches to matrix sparsification.

4.1.3 Graph Neural Networks

In our quest to further optimize the Quadratic Unconstrained Binary Optimization (QUBO) framework, we delve into the innovative domain of graph neural networks (GNNs), heralding a departure from conventional heuristic or mathematical techniques. GNNs represent a paradigm shift towards data-driven optimization, leveraging the predictive prowess of machine learning to enhance matrix sparsity. Our focus lies in harnessing the transformative potential of GNNs to learn optimal edge sparsification patterns [16] from richly annotated data sourced from Google OR-Tools. By immersing GNN models in a wealth of tour problems and their corresponding optimal solutions, we aim to equip these networks with the ability to discern and replicate the underlying structural nuances contributing to efficient edge sparsification. Through iterative learning iterations, these GNNs stand poised to unravel intricate patterns and dependencies intrinsic to the tour problem landscape, empowering them to make discerning decisions regarding edge selection and retention.

Central to this endeavor is the integration of node and edge embeddings, serving as the foundational building blocks for GNNs' predictive capabilities. Node embeddings encapsulate the latent features and attributes of each graph node, providing crucial contextual information for subsequent processing. Similarly, edge embeddings distill essential edge-level characteristics, facilitating nuanced understanding of inter-node relationships and connectivity patterns. By encoding these embeddings into the input layer of our neural networks, we enable GNNs to operate on a richly structured representation of the tour problem, fostering more informed decision-making regarding edge sparsification.

The journey towards GNN-based sparsification epitomizes our commitment to pushing the boundaries of optimization excellence through cutting-edge technologies. By harnessing the collective intelligence embedded within tour problem datasets, we seek to unlock new frontiers in matrix sparsity while preserving solution integrity and computational efficiency. Through the fusion of machine learning and optimization, we pave the way for transformative advancements in QUBO optimization, laying the groundwork for a future where data-driven insights drive unprecedented optimization capabilities.

4.1.3.1 Node Embeddings

Node embeddings serve as a critical component in our quest to imbue GNNs with the ability to understand and navigate the intricate structural dynamics of tour problems. By encoding each node's spatial coordinates into a feature-rich representation, we enable GNNs to capture and leverage spatial relationships inherent in the tour problem domain. These embeddings encapsulate essential geometric properties and topological insights, facilitating the network's understanding of node interconnectivity and proximity within the problem graph.

Moreover, the utilization of x and y coordinates as node features offers a unique advantage as in Figure 4.3, as it provides a natural and intuitive representation of node positions within the Euclidean space. This spatial information is instrumental in informing the network about the underlying geometry of the tour problem, guiding its decision-making process regarding edge sparsification. By embedding nodes into a high-dimensional space enriched with spatial context, we empower GNNs to glean actionable insights from geometric relationships, thereby enhancing their efficacy in edge sparsification tasks.

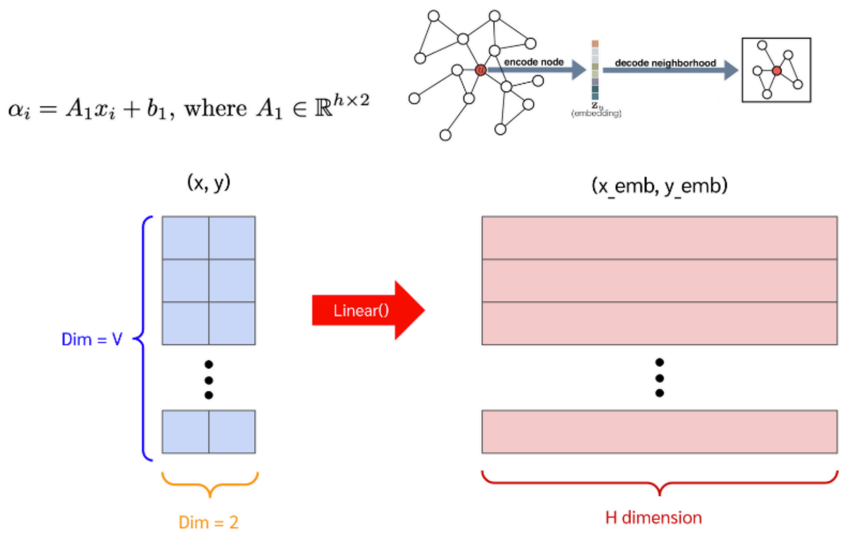


Figure 4.3 Embed node in Euclidean space into embedding space.

In the process of node embedding for our Quadratic Unconstrained Binary Optimization (QUBO) framework, we define the embedding function as follows:

$$\alpha_i = \mathcal{A}_1 x_i + b_1 \text{ where } \mathcal{A}_1 \in \mathbb{R}^{h \times 2}$$

- Here, α_i represents the embedded representation of node i, x_i denotes the feature vector corresponding to node i, and $\mathbb{A}$ represents the weight matrix mapping the input feature space to the embedding space. Additionally, b_1 is the bias term added to the transformation.

- This equation encapsulates the essence of node embedding by transforming the input node features x_i, into a higher-dimensional embedding space represented by α_i. The weight matrix $\mathcal{A}$, typically of size $h \times 2$, where h denotes the dimensionality of the embedding space, governs the transformation process, while the bias term b_1 provides an additional degree of freedom to the transformation.

4.1.3.2 Edge Embeddings

Edge embeddings serve as a cornerstone in our pursuit of enhancing QUBO optimization through machine learning-driven sparsification techniques. By encoding essential characteristics and relationships associated with problem edges into rich feature representations, we empower the GNNs to navigate the complex topology of tour problems with precision and efficacy.

4.1.3.2.1 Distance Matrix Embeddings The distance matrix provides valuable insights into the spatial arrangement of nodes within the problem graph, offering a quantitative measure of their pairwise distances or similarities. By encoding this distance information into edge embeddings, we equip graph neural networks (GNNs) with the ability to discern and exploit the spatial relationships embedded within the graph structure. This facilitates the learning of meaningful representations that capture the underlying graph topology in Figure 4.4, thereby facilitating the sparsification process and enhancing the efficiency and efficacy of QUBO optimization algorithms, we define the embedding function as follows:

$$\beta_{i,j} = \mathcal{A}_2 d_{i,j} + b_2 \text{ where } \mathcal{A}_2 \in \mathbb{R}^{\frac{h}{2}\times 1}$$

- Here, $\beta_{i,j}$ represents the embedded representation of the edge between nodes i and j, $d_{i,j}$ denotes the distance or similarity measure between nodes i and j, and $\mathcal{A}_2$ represents the weight matrix mapping the distance information to the embedding space. Additionally, b_2 is the bias term added to the transformation.

- This equation captures the essence of edge embedding by transforming the distance information between nodes into a higher-dimensional embedding space represented by $\beta_{i,j}$. The weight matrix $\mathcal{A}_{\in}$, typically of size $\frac{h}{2}\times 1$, where h denotes the dimensionality of the embedding space, governs the transformation process, while the bias term b_2 provides an additional degree of freedom to the transformation.

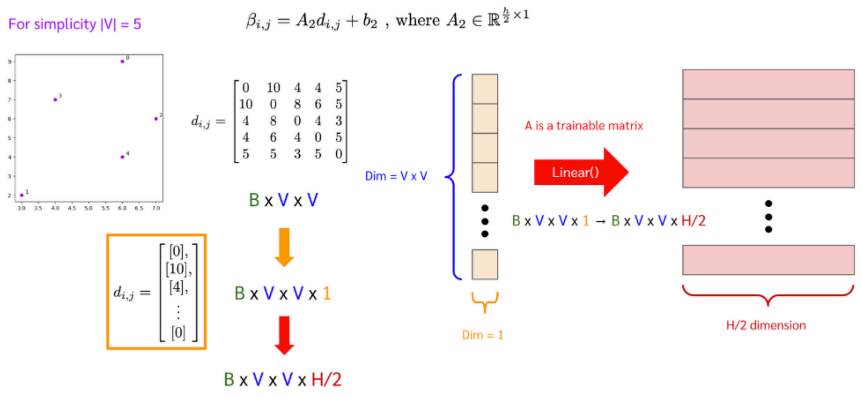

For simplicity |V| = 5
$\beta_{i,j} = A_2 d_{i,j} + b_2$, where $A_2 \in \mathbb{R}^{\frac{h}{2}\times 1}$
$d_{i,j} = \begin{bmatrix} 0 & 10 & 4 & 4 & 5 \\ 10 & 0 & 8 & 6 & 5 \\ 4 & 8 & 0 & 4 & 3 \\ 4 & 6 & 4 & 0 & 5 \\ 5 & 5 & 3 & 5 & 0 \end{bmatrix}$
B x V x V
$d_{i,j} = \begin{bmatrix} [0], \\ [10], \\ [4], \\ \vdots \\ [0] \end{bmatrix}$
B x V x V x 1
B x V x V x H/2
Dim = V x V
Dim = 1
A is a trainable matrix
Linear()
B x V x V x 1 → B x V x V x H/2
H/2 dimension

Figure 4.4 Embed distance information because it is crucial to the problem.

4.1.3.2.2 Neighborhood Embeddings

In addition to leveraging distance matrix information for edge embeddings, we also recognize the significance of incorporating neighborhood information to enhance the representational power of our graph neural network (GNN) model. By considering the k-nearest neighbors of each point of interest (POI) node, we aim to capture local structural characteristics and relationships within the graph topology in Figure 4.5. This inclusion of neighborhood information enables our GNN to discern and exploit spatial dependencies and correlations between adjacent nodes, thereby enriching the learned embeddings with contextual insights. Moreover, by parameterizing the value of k, we afford flexibility to the model, allowing users to fine-tune the level of granularity in the neighborhood representation. Users can adjust the value of k based on the specific characteristics of the graph structure and the desired level of detail in capturing local connectivity patterns. Consequently, by integrating both distance matrix information and neighborhood relationships into our edge embedding process, we equip our GNN model with a comprehensive understanding of the graph topology, enabling it to generate robust and informative representations conducive to effective sparsification and optimization.

Incorporating the k-nearest neighbor (kNN) information into our edge embeddings, we introduce the equation:

$$\beta_{i,j} = \mathcal{A}_3 \delta_{i,j}^{kNN} \text{ where } \mathcal{A}_3 \in \mathbb{R}^{\frac{h}{2}\times 1}, \delta_{i,j}^{kNN} \in \{0,1,2\}$$

- Here, $\mathcal{A}_3$ represents the weight matrix associated with the kNN distance information, with dimensions $\frac{h}{2} \times 1$. The term $\delta_{i,j}^{kNN}$ denotes the kNN distance indicator, which takes on values in the set $\{0, 1, 2\}$

to signify the absence of a kNN relationship (0), a node j be a kNN relationship of node i (1), or an edge (i, j) is a self-connection to itself (2). By integrating this kNN information into our edge embeddings, we augment the representation with local connectivity insights, allowing the graph neural network (GNN) model to capture fine-grained structural dependencies within the graph topology. This enhancement facilitates more nuanced and contextually rich embeddings, empowering the model to make informed decisions regarding edge selection and retention during the sparsification process.

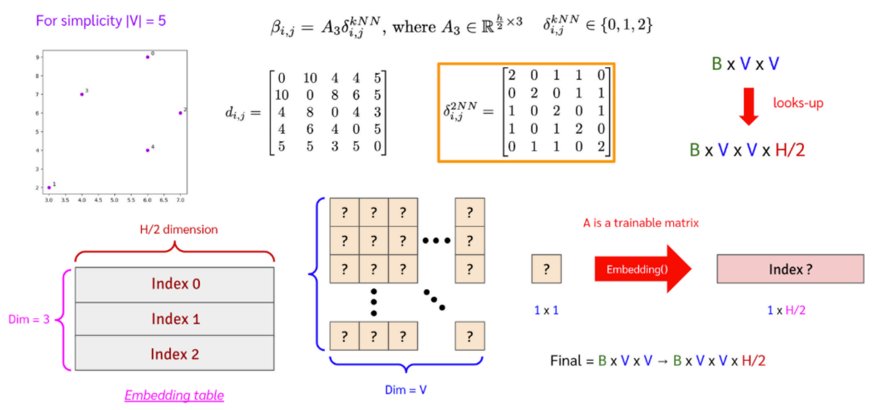

For simplicity |V| = 5
$\beta_{i,j} = A_3\delta^{kNN}_{i,j}$, where $A_3 \in \mathbb{R}^{\frac{h}{2}\times 3}$ $\delta^{kNN}_{i,j} \in \{0,1,2\}$
$d_{i,j} = \begin{bmatrix} 0 & 10 & 4 & 4 & 5 \\ 10 & 0 & 8 & 6 & 5 \\ 4 & 8 & 0 & 4 & 3 \\ 4 & 6 & 4 & 0 & 5 \\ 5 & 5 & 3 & 5 & 0 \end{bmatrix}$
$\delta^{2NN}_{i,j} = \begin{bmatrix} 2 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 \\ 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 & 0 \\ 0 & 1 & 1 & 0 & 2 \end{bmatrix}$
B x V x V
looks-up
B x V x V x H/2
H/2 dimension
Dim = 3
Index 0
Index 1
Index 2
Embedding table
Dim = V
A is a trainable matrix
Embedding()
Index ?
1 x 1
1 x H/2
Final = B x V x V → B x V x V x H/2

Figure 4.5 Embed neighborhood information so the model can understand the local structure of the graph.

4.1.3.2.3 Combine Edge Embeddings Combining edge embeddings in Figure 4.6 derived from both distance matrix information and k-nearest neighbor (kNN) information serves to enrich the representation of graph topology in our graph neural network (GNN) model. The inclusion of distance matrix information allows us to capture global geometric relationships between nodes, providing insights into the overall layout and spatial arrangement of the graph. On the other hand, incorporating kNN information enables us to capture local connectivity patterns, highlighting the nearest neighbors of each node and their corresponding distances.

By combining these two sources of information, we create a more comprehensive and nuanced representation of the graph structure. The edge embeddings derived from distance matrix information offer insights into the overall spatial distribution of nodes and their pairwise distances, while the kNN-based embeddings capture finer-grained details about local neighborhoods and connectivity patterns. This combined representation allows our GNN model to leverage diverse information sources, providing a richer understanding of the underlying graph topology.

Furthermore, we enable the GNN model to capture complex structural relationships in the graph by including both global and local information into the edge embeddings. The model is better equipped to balance local connection limitations with global geometric concerns throughout the sparsification process because to this holistic viewpoint. Consequently, the amalgamated edge embeddings enable increasingly efficient and resilient sparsification tactics, thereby augmenting the efficacy and productivity of our optimization structure.

$$\beta_{i,j} = \mathcal{A}_2 d_{i,j} + b_2 \mid\mid \mathcal{A}_3\delta^{kNN}_{i,j}$$

Concatenation is indeed a widely used technique in machine learning, particularly in the context of feature representation and model architecture design. In machine learning, concatenation refers to the process of combining multiple feature vectors or representations into a single, larger feature vector. This technique is often employed to merge different types of information or representations, allowing models to leverage

$$\beta_{i,j} = A_2 d_{i,j} + b_2 \| A_3 \delta_{i,j}^{kNN}, \text{ where } A_2 \in \mathbb{R}^{\frac{h}{2}\times 1}, A_3 \in \mathbb{R}^{\frac{h}{2}\times 3}$$

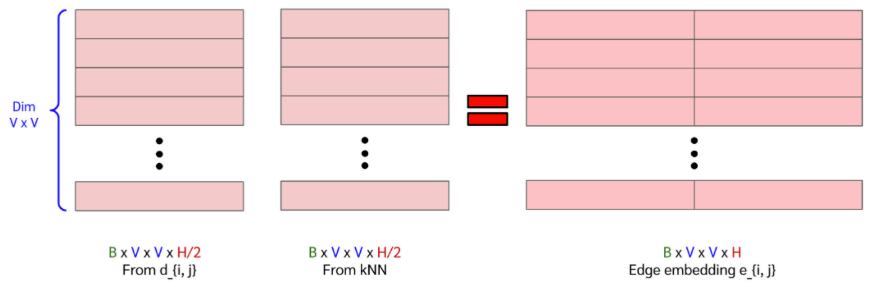


Figure 4.6 Concatenate distance and neighborhood information into one edge embedding information.

diverse sources of data or features for improved performance. For example, in neural network architectures, concatenation can be used to combine embeddings derived from different modalities or sources of information, enabling the model to learn from multiple data domains simultaneously. Additionally, concatenation can be used in tasks such as sequence modeling or natural language processing, where it facilitates the combination of features extracted from different parts of the input data. Overall, concatenation serves as a powerful tool in machine learning, enabling models to integrate and leverage heterogeneous information for more effective learning and inference.

4.1.3.3 Edge Features

The edge features are computed using a combination of learned weight matrices W_1 and W_2, along with the node embeddings x_i and x_j. These features capture both the intrinsic properties of the edge itself ($e_{i,j}$) as well as the attributes of the nodes connected by the edge (x_i and x_j). By combining these elements, we create a comprehensive representation of the edge that incorporates both local and global information from the graph.

$$W_1 e_{i,j} + W_2(x_i + x_j)$$

The first component of the edge features, $W_1 e_{i,j}$, focuses on the intrinsic characteristics of the edge, which may include properties such as edge weight, distance, or similarity. The weight matrix W_1 is learned during the training process and is used to transform the raw edge information $e_{i,j}$ into a higher-dimensional representation that captures relevant features.

The second component, $W_2(x_i + x_j)$, leverages the node embeddings x_i and x_j to incorporate information about the nodes connected by the edge. By summing the embeddings of the two nodes and applying another learned weight matrix W_2, we create a combined node representation that reflects the attributes of both nodes involved in the edge.

Overall, the edge features capture a rich set of information about the relationships between nodes in the graph, encompassing both local edge properties and global node attributes. This comprehensive representation enables our model to effectively learn the underlying structure of the graph and make informed decisions during the optimization process.

4.1.3.4 Node Features

The node features are computed using a combination of learned weight matrices W_3 and W_4, along with the node embeddings x_i. These features capture both the intrinsic properties of the node itself as well as the attributes of its neighboring nodes. By combining these elements, we create a comprehensive representation of the node that incorporates both local and global information from the graph.

$$W_3 x_i + \sum_{j \in \mathcal{N}(i)} W_4 x_j$$

The first component of the node features, $W_3 x_i$, focuses on the intrinsic characteristics of the node. The weight matrix W_3 is learned during the training process and is used to transform the raw node embedding x_i into a higher-dimensional representation that captures relevant features.

The second component, $\sum_{j \in \mathcal{N}(i)} W_4 x_j$, leverages the embeddings of the neighboring nodes x_j within the node's neighborhood $\mathcal{N}(i)$. By summing the embeddings of the neighboring nodes and applying another learned weight matrix W_4, we create a combined representation that reflects the attributes of both the node and its neighbors.

In summary, the node features capture a rich set of information about the node and its surrounding context within the graph, enabling our model to effectively learn the underlying structure and make informed decisions during the optimization process.

4.1.3.5 Graph Convolutional Layers

The graph convolutional layers (GCN) play a crucial role in aggregating information from the edges and nodes of the graph to generate meaningful feature representations for further processing. In the context of our optimization framework, these layers facilitate the extraction of high-level features that capture the structural properties and relationships within the graph.

$$e_{i,j}^{\ell=0} = \beta_{i,j}$$

The edge feature information is computed iteratively across multiple layers (ℓ) of the GCN. At the initial layer ($\ell = 0$), the edge features $e_{i,j}^{\ell=0}$ are initialized with the pre-computed edge embeddings $\beta_{i,j}$. Subsequently, in each subsequent layer, the edge features are updated using a combination of learned weight matrices W_1^ℓ and W_2^ℓ, along with the node embeddings x_i^ℓ and x_j^ℓ. This process involves applying a series of operations, including matrix multiplication, batch normalization (BN), rectified linear unit (ReLU) activation, and element-wise addition.

$$e_{i,j}^{\ell+1} = e_{i,j}^{\ell} + ReLU(BN(W_1^\ell e_{i,j}^\ell + W_2^\ell x_i^\ell + W_2^\ell x_j^\ell))$$

Similarly, the node feature information is computed across multiple layers (ℓ) of the GCN. At the initial layer ($\ell = 0$), the node features $x_i^{\ell=0}$ are initialized with the pre-computed node embeddings α_i. In each subsequent layer, the node features are updated using a combination of learned weight matrices W_3^ℓ and W_4^ℓ, along with the embeddings of neighboring nodes x_j^ℓ. Additionally, a normalization term $\eta_{i,j}^\ell$ is applied to adjust the contribution of each neighboring node based on the edge features $e_{i,j}^\ell$.

$$x_i^{\ell=0} = \alpha_i$$

Epsilon (ϵ) is a very small number to avoid division by zero of $\eta_{i,j}$ neighborhood aggregation from its edge connections.

$$x_i^{\ell+1} = x_i^\ell + ReLU(BN(W_3^\ell x_i^\ell + \sum_{j \in \mathcal{N}(i)} W_4^\ell x_j^\ell \odot \eta_{i,j}^\ell)) \text{ where } \eta_{i,j}^\ell \frac{\sigma(e_{i,j}^\ell)}{\sum_{j' \in \mathcal{N}(i)} \sigma(e_{i,j'}^\ell) + \epsilon}$$

The GCN takes as input the edge features $e_{i,j}^\ell$ and node features x_i^ℓ at each layer (ℓ) and processes them to generate updated edge and node features $e_{i,j}^{\ell+1}$ and $x_i^{\ell+1}$. This process is repeated for multiple layers (ℓ)

until the desired level of feature extraction is achieved. The output of the last layer ($\ell = L$) represents the final feature representations that capture the essential characteristics of the graph.

$$e_{i,j}^{\ell+1}, x_i^{\ell+1} = GCN(e_{i,j}^{\ell}, x_i^{\ell}) \; \forall \; \ell \in L$$

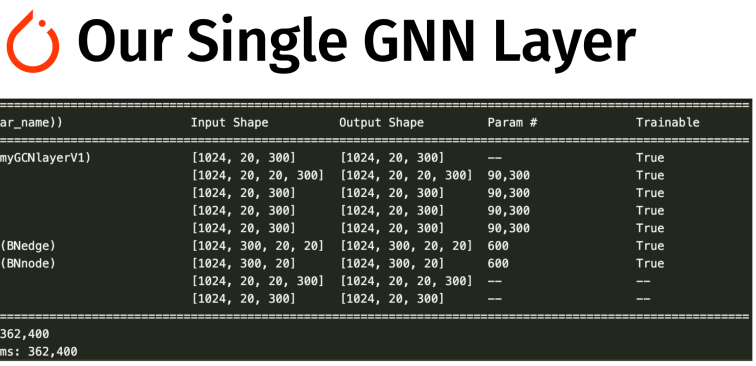
Finally, the output of the last layer of the GCN is fed into a multi-layer perceptron (MLP) to predict the optimal TSP route. The MLP takes the edge features $e_{i,j}^{L}$ as input and generates the predicted probabilities $p_{i,j}^{TSP}$ for each edge in the graph. These probabilities represent the likelihood of including each edge in the optimal TSP route, allowing for efficient and effective route planning.

$$p_{i,j}^{TSP} = MLP(e_{i,j}^{L})$$

4.1.4 Implementation in PyTorch

Incorporating PyTorch for the implementation of our optimization framework offers several advantages owing to its versatility, efficiency, and extensive library support. PyTorch is a widely adopted deep learning framework known for its flexibility and ease of use, making it an ideal choice for developing complex neural network architectures, such as graph convolutional networks (GCNs), and training them on large-scale datasets.

One of the key benefits of PyTorch is its dynamic computational graph construction, which allows for intuitive model development and debugging. Unlike static graph frameworks, where the computational graph must be predefined before executing the model, PyTorch constructs the graph on-the-fly during execution, enabling more dynamic and flexible model designs. This dynamic nature of PyTorch simplifies the process of experimenting with different network architectures and hyperparameters, facilitating rapid prototyping and iteration. Additionally, PyTorch provides seamless integration with GPU acceleration, leveraging the computational power of modern graphics processing units (GPUs) to accelerate training and inference tasks significantly. By harnessing the parallel computing capabilities of GPUs, PyTorch enables faster model training and inference, essential for handling large-scale optimization problems efficiently.



```
==========================================================================================
Layer (type (var_name))              Input Shape          Output Shape         Param #              Trainable
==========================================================================================
myGCNlayerV1 (myGCNlayerV1)          [1024, 20, 300]      [1024, 20, 300]      --                   True
├─Linear (W1)                        [1024, 20, 20, 300]  [1024, 20, 20, 300]  90,300               True
├─Linear (W2)                        [1024, 20, 300]      [1024, 20, 300]      90,300               True
├─Linear (W3)                        [1024, 20, 300]      [1024, 20, 300]      90,300               True
├─Linear (W4)                        [1024, 20, 300]      [1024, 20, 300]      90,300               True
├─BatchNorm2d (BNedge)               [1024, 300, 20, 20]  [1024, 300, 20, 20]  600                  True
├─BatchNorm1d (BNnode)               [1024, 300, 20]      [1024, 300, 20]      600                  True
├─ReLU (ReLU)                        [1024, 20, 20, 300]  [1024, 20, 20, 300]  --                   --
├─ReLU (ReLU)                        [1024, 20, 300]      [1024, 20, 300]      --                   --
==========================================================================================
Total params: 362,400
Trainable params: 362,400
```

Figure 4.7 Summary of our model architecture in one GNN layer.

The torchinfo summary provides a concise overview of our model's architecture as you can see from Figure 4.7, detailing key attributes such as layer types, input and output shapes, parameter counts, and computational complexities. This summary aids in understanding how data flows through the network and helps identify areas for optimization. It serves as a valuable tool for validating the model's configuration and ensuring its effectiveness.

4.1.5 Initial Model Predictions

In this phase, we randomly generate 20 sets of TSP points and feed them into our trained model, which has been supervised using data from Google OR-Tools, renowned for its efficiency in solving TSP. The model's predictions are visualized as a heatmap representing the probability that each edge occurs in the actual TSP route. This approach allows us to assess the model's performance in predicting edge occurrences and provides insights into its ability to generate optimal TSP solutions based on the learned patterns from the supervised data.

4.1.5.1 Loss curves

In our training process, we utilize the Negative Log Likelihood (NLL) loss function to evaluate the performance of our model. The NLL loss function is commonly employed in classification tasks and measures the difference between the predicted probabilities and the true labels. By minimizing the NLL loss, our model aims to accurately predict the likelihood of each edge occurrence in the TSP route. Additionally, we employ the Adam optimizer during the training process. Adam is an adaptive learning rate optimization algorithm that efficiently updates model parameters based on the gradients of the loss function. By adaptively adjusting the learning rates for each parameter, Adam helps accelerate the convergence of the training process and enhances the overall performance of the model.

In our experimental setup, we partition our dataset into three distinct subsets: training, testing, and validation as shown in Figure 4.8. The training and testing sets are utilized for the conventional purposes of model training and evaluation, respectively, as commonly practiced in machine learning tasks. However, our validation set serves a specific purpose in our training pipeline. It is employed for monitoring the model's performance and scheduling the learning rate throughout the training epochs. By assessing the model's performance on the validation set at regular intervals during training, we can make informed decisions about adjusting the learning rate to optimize training convergence and prevent overfitting. This strategic use of the validation set ensures that our model generalizes well to unseen data and achieves robust performance across different epochs.

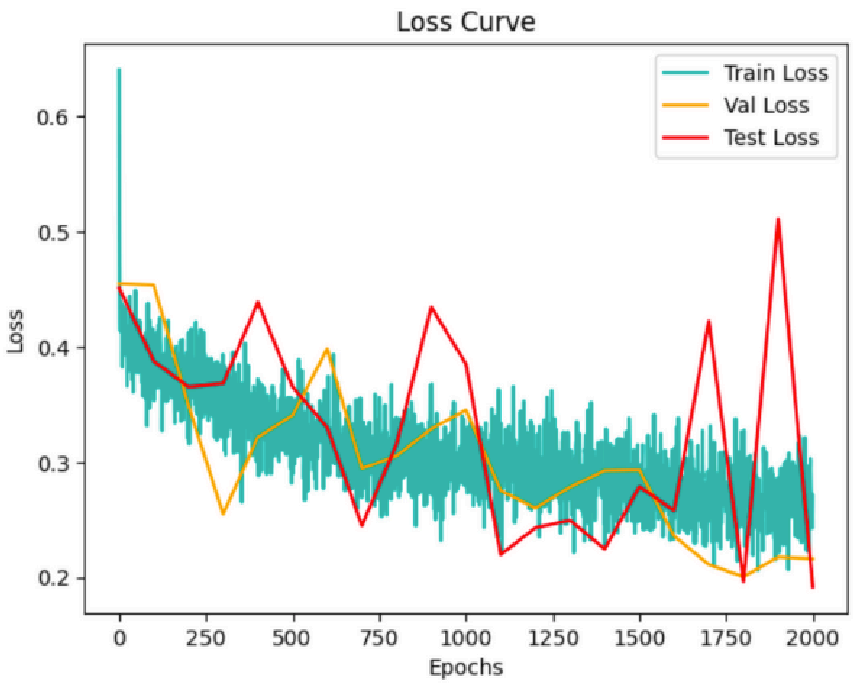


Figure 4.8 Our first model training loss curves (without any well-fine-tuning procedure).

4.1.5.2 Heatmap Visualization

In the Heatmap Visualization section, we present the initial results of our model. The term *heatmap* refers to the graphical representation of edge probabilities generated by our model. Each cell in the heatmap corresponds to a pair of nodes in the TSP problem, with the color intensity indicating the probability assigned by the model to the existence of an edge between those nodes. As depicted in Figure 4.9, the heatmap illustrates the model's predictions regarding the likelihood of edges occurring in the optimal TSP route.

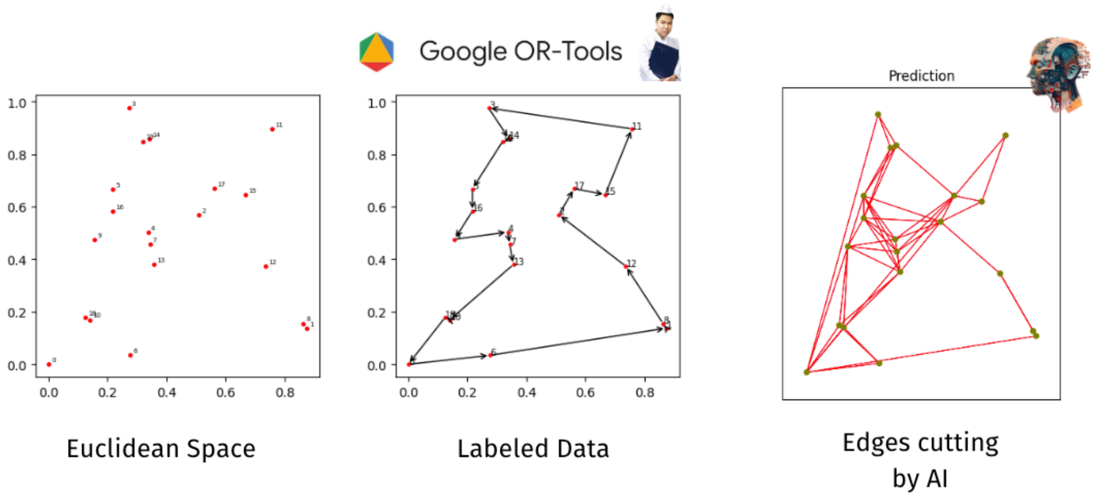


Figure 4.9 This is the experimental result from previous loss curves, with 20 random points.

We observe a mix of correct predictions alongside some missed edges that are present in the label data. Despite these discrepancies, it's crucial to note that the model demonstrates an ability to discern patterns and make informed predictions. This early indication suggests that our approach holds promise as a viable technique for matrix sparsification in the pre-computational phase. As we continue to refine and fine-tune the model, we anticipate further improvements in prediction accuracy and the overall efficacy of our sparsification strategy.

4.2 Computational: D'Wave state of the art quantum machine

The distance matrices generated during the pre-computational phase for the modified Traveling Salesman Problem instances were subsequently utilized as input data for the D-Wave quantum annealing computer. The objective function values, representing the respective tour costs, were obtained as a result of these quantum computations. These computed tour costs will be compared against the optimal tour costs provided by the TSPLIB library instances. This comparative analysis aims to evaluate the efficacy and potential benefits accrued through the pre-computational processing procedures employed before leveraging the quantum annealing approach.

In this experiment, two problems from TSPLIB [17] were chosen as the main focus; GR17 contains 17 nodes of traversal and FRI26 contains 26 nodes of traversal. The modified instances of the distance matrices have been categorized into three primary groups: those generated by deep neural networks, greedy algorithms, and graph spanners. These problem instances will be subjected to computation on the D-Wave Binary Quadratic Model (BQM) solver, with time limits set at 10, 30, 60, and 120 seconds, respectively. This systematic approach aims to investigate the relationship between the computational resource consumption and the quality of the solutions obtained, thereby elucidating the trade-offs involved in the solution process.

4.2.1 Greedy Algorithm Results

The results obtained from the distance matrices modified by greedy algorithms exhibit highly unpredictable and intriguing characteristics. Notably, no guarantee allocating additional computational time will yield improved solutions. However, an interesting observation is that the cost of traversal gradually decreases as the sparsity of the matrices increases. This anomalous behavior warrants further investigation to ascertain whether it is merely a coincidental occurrence or if there are underlying technical reasons that can explain this phenomenon. A thorough examination of the greedy algorithms' mechanisms and their interactions with the matrix sparsity is necessary to elucidate the observed behavior and determine its implications for the solution quality and computational efficiency. Such an investigation may uncover novel insights or principles that inform the development of more robust and efficient optimization techniques, particularly in the context of sparse or structured problem instances.

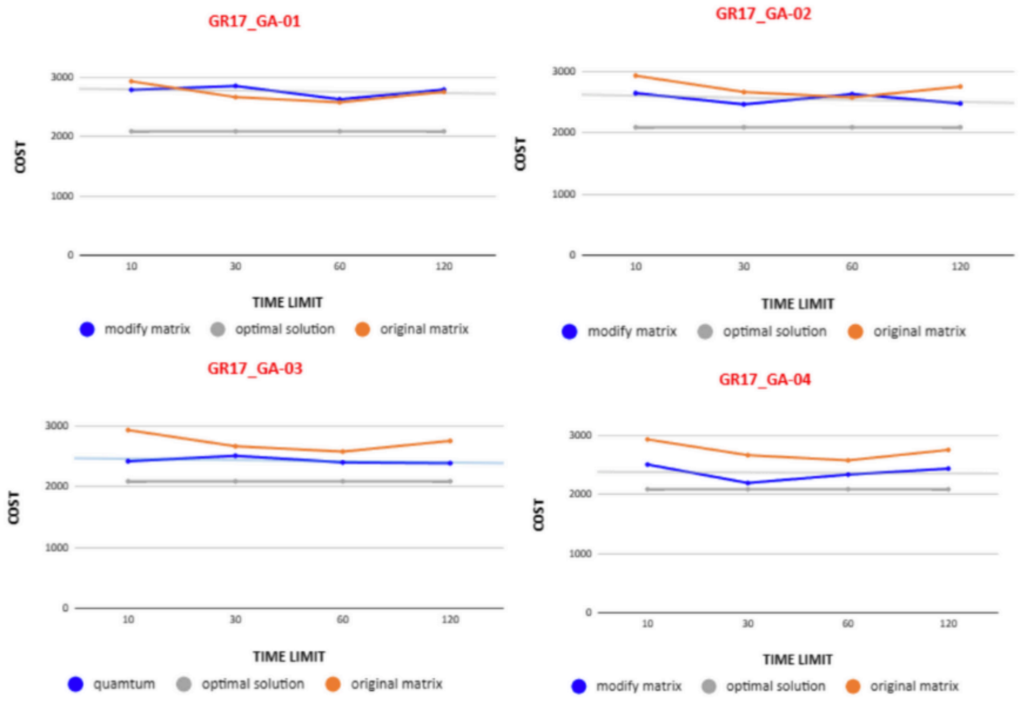


Figure 4.10 Line graph illustrates the cost of traversal of the distance matrix of problem GR17 modified by greedy algorithms with sparsity GA-01 5.8%, GA-02 10%, GA-03 13.5%, and GA-04 17% at time limits of 10, 30, 60, 120 seconds.

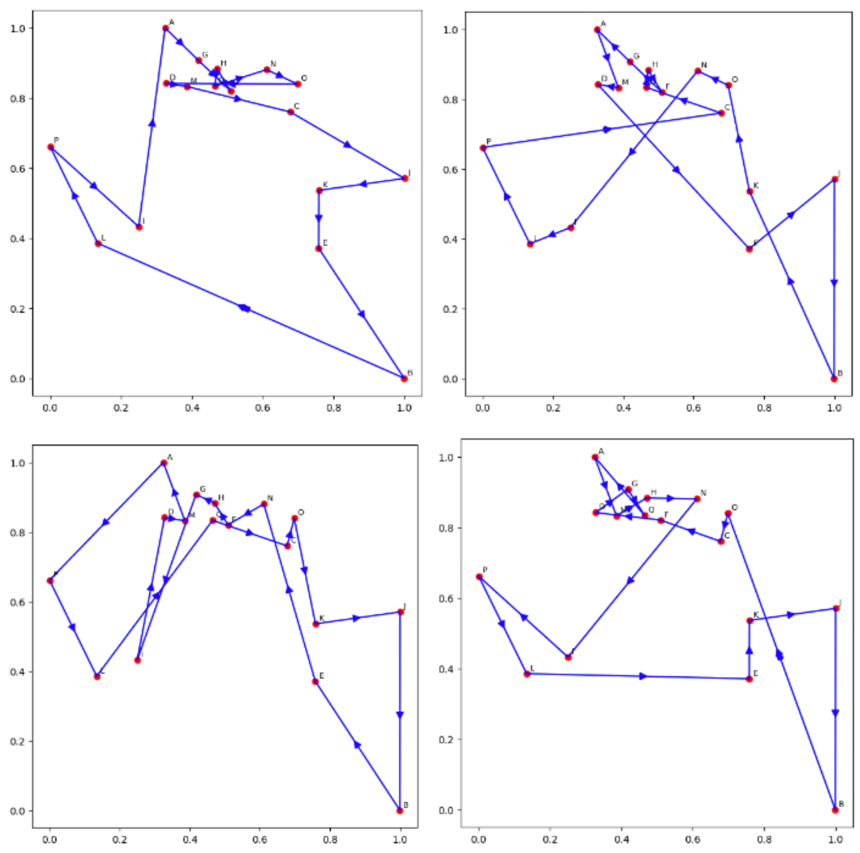


Figure 4.11 Traversal Graph plotted a solution generated from modified matrix GR17 GA-01 with a time limit of 10, 30, 60, and 120 seconds.

4.2.2 Graph Theory (Graph Spanner) Results

The results obtained from the distance matrices modified by the graph spanner approach have proven to be the least favorable among all the pre-computation techniques employed. Not only are the results highly unpredictable, but they also frequently yield invalid solutions that violate the constraints of the problem, rendering them inadmissible for statistical analysis. The underlying cause of this issue is conjectured to stem from the fundamental principle of graph spanners, which selects subgraphs that preserve certain characteristics of the original graph. However, this approach may not be well-suited for the Traveling Salesman Problem (TSP), as the process of constructing a spanner could potentially disrupt crucial traversal paths within the graph.

The inherent trade-off between preserving specific graph properties and maintaining the integrity of all feasible traversal routes appears to be a critical factor contributing to the suboptimal performance of the graph spanner method in this context. Further investigation is warranted to ascertain the precise reasons behind this phenomenon and to explore potential modifications or alternative techniques that could mitigate these issues while retaining the desirable properties of graph spanners. Such endeavors may yield valuable insights into the intricate interplay between graph theory and combinatorial optimization problems, potentially paving the way for the development of more robust and effective pre-computation strategies tailored to the unique challenges of the TSP and related domains.

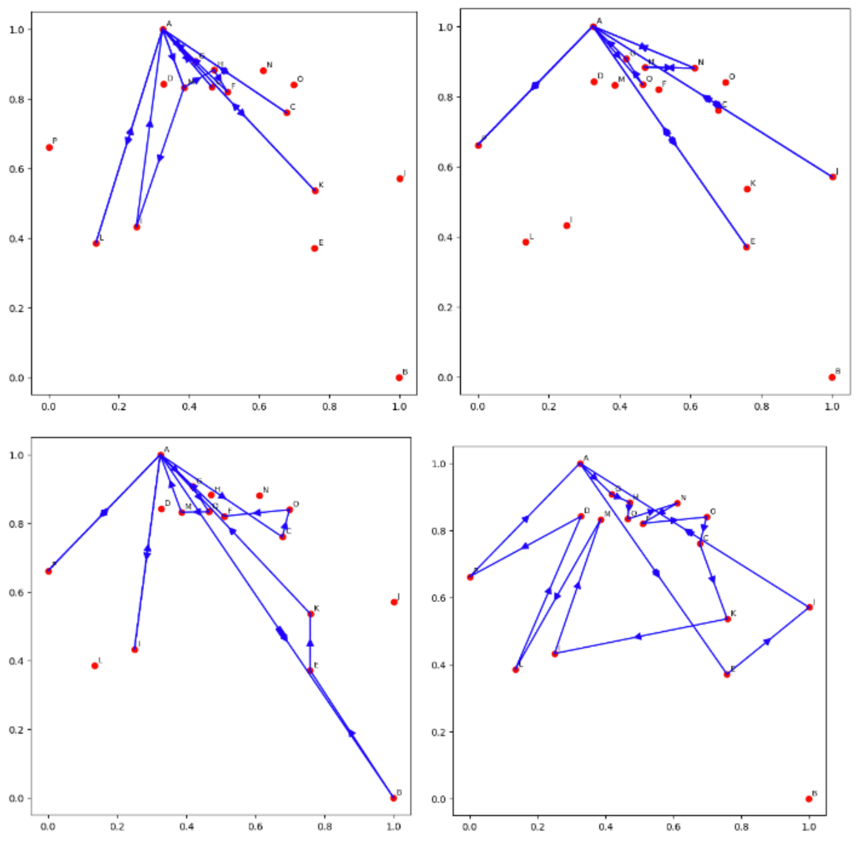


Figure 4.12 Traversal Graph plotted a solution generated from modified matrix GR17 SP-INF t105 with a time limit of 10, 30, 60, and 120 seconds that gives all false solutions.

4.2.3 Deep Neural Networks Results

The distance matrices modified by the Deep Neural Network approach have exhibited a remarkable and intriguing characteristic, as the solutions obtained remain consistent regardless of the imposed time limits. This observation indicates that, if further examination corroborates these findings and the Deep Neural Network method consistently generates invariant results, it would confer a significant advantage. Specifically, this approach would require minimal computational resources to produce a particular solution, thereby offering substantial efficiency gains. The potential to generate high-quality solutions with minimal resource consumption is a highly desirable trait, especially in resource-constrained environments or when dealing with large-scale or complex problem instances. Consequently, a comprehensive investigation into the Deep Neural Network method and its underlying principles is warranted, as it holds the promise of enabling efficient and scalable solutions to complex optimization problems while minimizing the associated computational overhead.

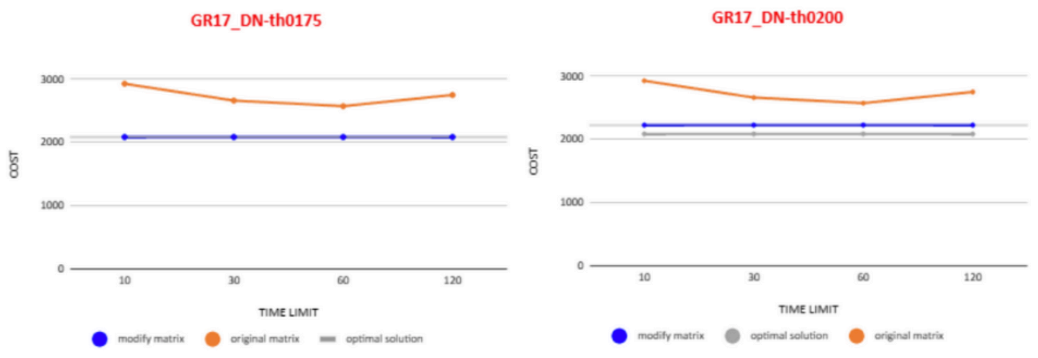


Figure 4.13 Line graph illustrates the cost of traversal of the distance matrix of problem GR17 modified by the deep neural network at a time limit of 10, 30, 60, and 120 seconds. GR17 DN-th0175 gives the cost of traversal of 2085 which is equal to the optimal solution traversal cost provided by TSPLIB and GR17 DN-th0200 which gives the cost of traversal 2226.

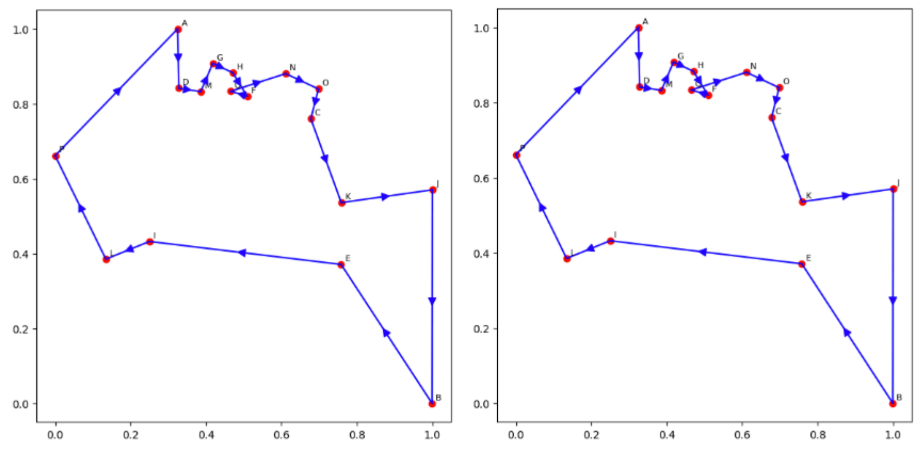


Figure 4.14 Traversal Graph plotted a solution generated from modified matrix GR17 DN-th0150 with a time limit of 10s (left) and 120s (right) produced an identical path and cost of traversal.

CHAPTER 5 CONCLUSIONS

Firstly, we begin by reporting the challenges encountered during our experiments. We found that the D-Wave quantum computer could not handle large-scale problems effectively. Therefore, our initial proposal to conduct experiments with small, medium, and large datasets has been revised. We will only experiment with smaller datasets, specifically GR17 and FRI26 from TSPLIB95.

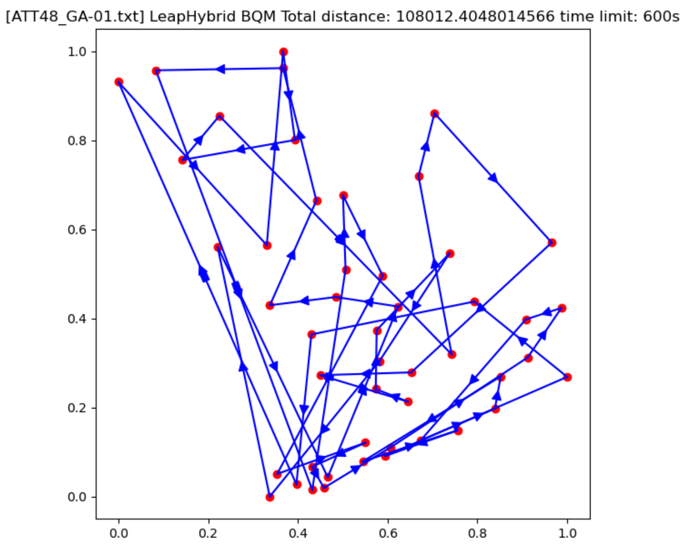


Figure 5.1 This image illustrates the experimental results indicating that ATT48 with a size $|V| = 48$ cannot perform well on the current D-Wave quantum hardware (using the free, non-commercial account).

Secondly, regarding our experiment, we aim to integrate quantum computing technology into real industrial applications. Therefore, we chose D-Wave, the first commercial quantum computer offering various products. Our focus is primarily on the Leap Hybrid Solver for our experiments. This is because other solvers, such as the 2000Q, perform poorly when the variable size is too large. For example, in our Traveling Salesperson Problem (TSP) with ten nodes, the 2000Q could not process the data due to insufficient qubits available or allowed for use, rendering it unable to solve this problem. Our project has faced numerous challenges and obstacles in conducting many experiments. However, we remained undeterred until we achieved successful experimentation and obtained satisfactory results under certain assumptions. Consequently, we are particularly eager to convey this aspiration to our readers. Nevertheless, due to the extensive details involved, we must carefully devise a method of presentation that effectively communicates our findings. In this discussion, we have chosen to highlight experimental results that we believe effectively demonstrate the objectives of our project. If you require further information or additional details, please get in touch with us for further clarification at your convenience.

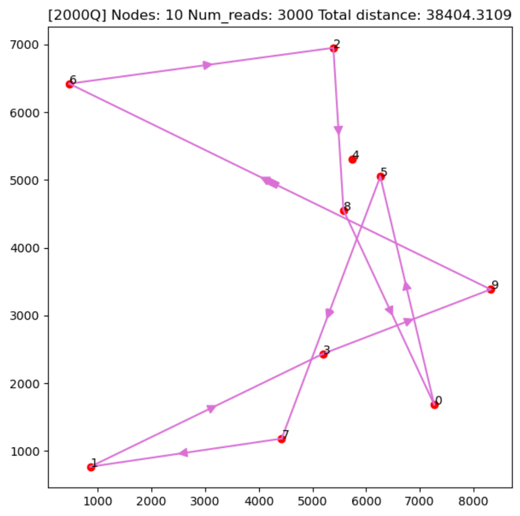


Figure 5.2 In this experiment, we randomly generated a ten-node problem to test on the D-Wave 2000Q quantum annealer. We found that it provided an unacceptable solution due to the constraints of the Hamiltonian cycle not being satisfied (some nodes were not visited) by the Quantum Annealer. Therefore, we choose not to endorse this solver for our current work.

5.1 Conclusion of Approach 1: Greedily Matrix Sparsification

Let's begin summarizing the results of our experiment, starting with our initial approach, which involved making the distance matrix of the TSP problem more sparse. We hypothesized that by increasing the sparsity of the matrix, the QUBO computation would become significantly more straightforward to solve. This approach was our first experimental method. The rationale behind this hypothesis lies in the observation that when a matrix becomes more sparse, the computational complexity associated with solving the QUBO formulation tends to decrease. By reducing the density of connections between nodes in the distance matrix, we aimed to simplify the computational process for quantum solvers, such as those utilized in D-Wave's quantum annealing approach. This initial experimentation is a foundational step in exploring techniques to enhance the efficiency and effectiveness of quantum computing algorithms for solving combinatorial optimization problems like the TSP.

Although this method might have seemed rudimentary initially, we deemed it necessary to pursue it because we aimed to establish a foundational framework for future algorithm development. During the experimentation phase, we established a threshold to measure the percentage sparsity of our matrix. This threshold served as a guideline for evaluating the effectiveness of our approach. The results of our experimentation, as depicted in Figure 5.3, demonstrate that we could improve the quality of solutions within the same time limit constraints as before. This suggests that our method can potentially mitigate the computational overhead of solving QUBO problems. This approach lays the groundwork for further advancements in optimizing quantum computing algorithms for real-world applications by effectively reducing computational complexity without sacrificing solution quality.

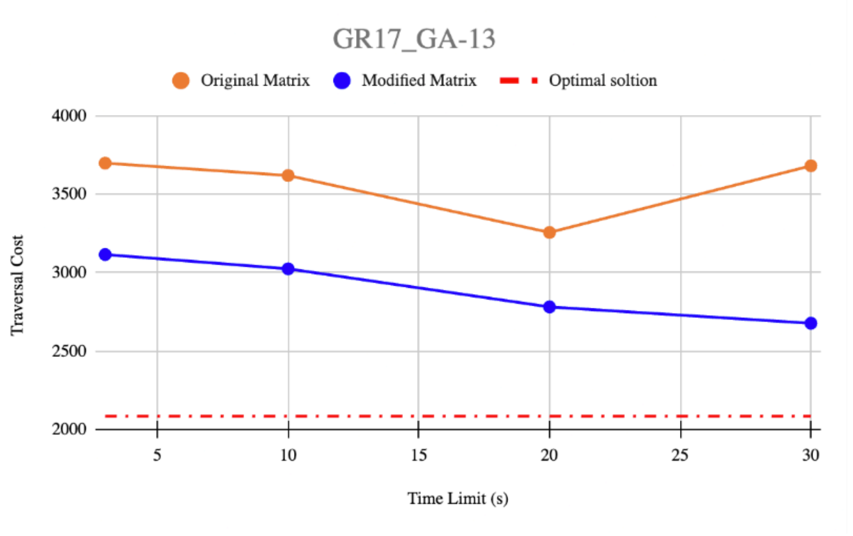


Figure 5.3 Our experimental findings revealed that despite the simplicity inherent in the matrix sparsification method, it surprisingly yielded significant improvements in the resulting cost efficiency.

However, there is an important observation to note. While we did achieve a reduction in the objective function value, upon visualizing the resulting paths, we noticed that they lacked coherence. In other words, the paths appeared nonsensical and impractical for real-world applications. If these paths were to be utilized in business operations, they might not be considered viable options. Instead, decision-makers may rely on their intuition to navigate routes, as the visual representation of the paths, although feasible, lacks aesthetic appeal and practicality. This observation highlights the importance of optimizing objective function values and ensuring that the solutions generated are intuitively viable and understandable for real-world implementation, as shown in Figure 5.4.

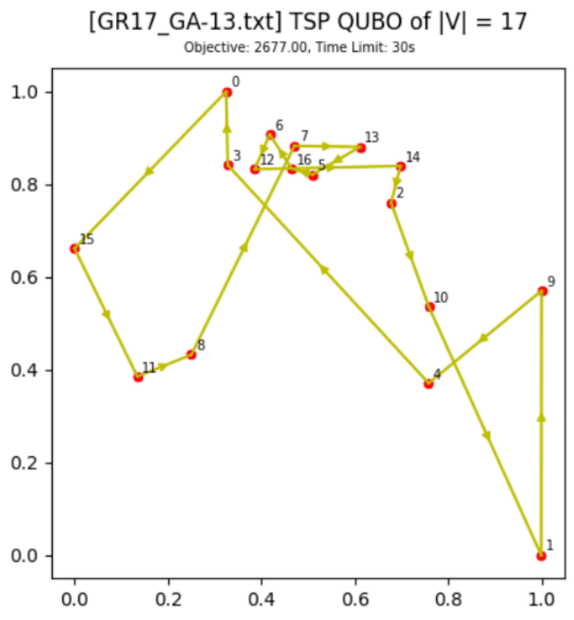


Figure 5.4 This is an example image depicting the experimental results of the Matrix Sparsification method, also referred to as the Greedy Approach. While the quality of the solutions has indeed improved compared to the original, it's worth noting that the resulting paths may not be driver-friendly.

5.2 Conclusion of Approach 2: Shortest Path Graph Spanner

Building upon the previous method, we discovered that increasing the sparsity of the matrix simplifies the problem and facilitates faster QUBO computations. However, it's crucial to consider what elements we sparseify. For instance, randomly sparsifying nodes without careful consideration may inadvertently distort specific characteristics of the TSP problem we aim to address. Suppose we blindly remove nodes or edges to reduce the complexity of the problem. In that case, it may inadvertently disrupt some of the critical features of the TSP problem we intend to solve. Consequently, the resulting solution may require further evaluation to determine its suitability for practical application. This emphasizes the need for thoughtful consideration when applying sparsification techniques, as they can significantly impact the quality and practicality of the solutions obtained.

Therefore, we are keen to approach this problem from the perspective of mathematicians. Throughout our research and efforts to understand the situation we faced, we have devised an idea to leverage graph theory to aid in simplifying this issue. Specifically, we have chosen to utilize Graph Spanner, which is based on the fundamental concept of creating a new graph while preserving the characteristics of the original graph as much as possible under certain conditions. In our experimentation, we have endeavored to construct a Spanner that represents the original problem by attempting to maintain the distances of the shortest paths in the new graph not to exceed a certain multiple, denoted by t, of the distances in the original graph. This variable t is a parameter we have investigated in this experimentation.

From the experimental results presented in Figure 5.5, we observed a significant reduction in the objective function values, particularly within certain time limits, compared to the original QUBO matrix. However, our interest extends beyond merely minimizing the objective function; we also focus on the practical usability of the resulting paths. Upon plotting the paths using multi-dimensional scaling in scikit-learn, which approximates the distances of the distance matrix onto Euclidean space, as shown in Figure 5.6, we found that the results were not entirely satisfactory. Despite improving the objective function values, the paths generated did not meet our expectations.

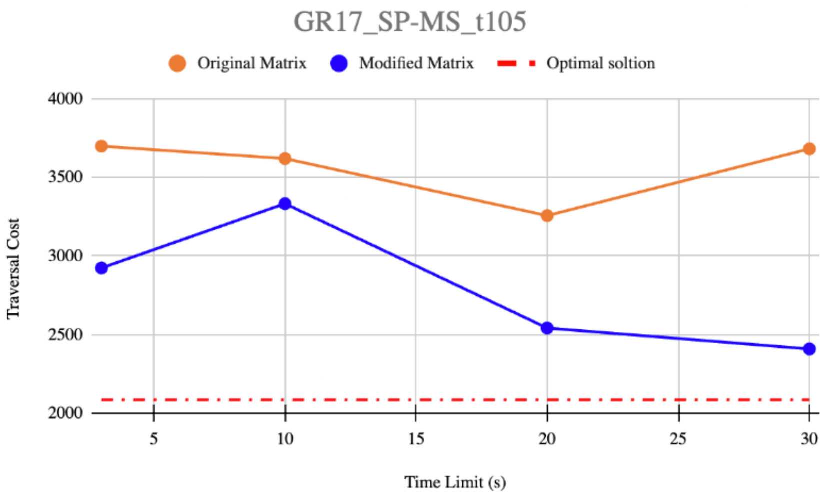


Figure 5.5 We have incorporated the path-greedy t spanner method to address this issue. Upon examining the objective function values, we found that this approach indeed improved the outcomes of our experiments.

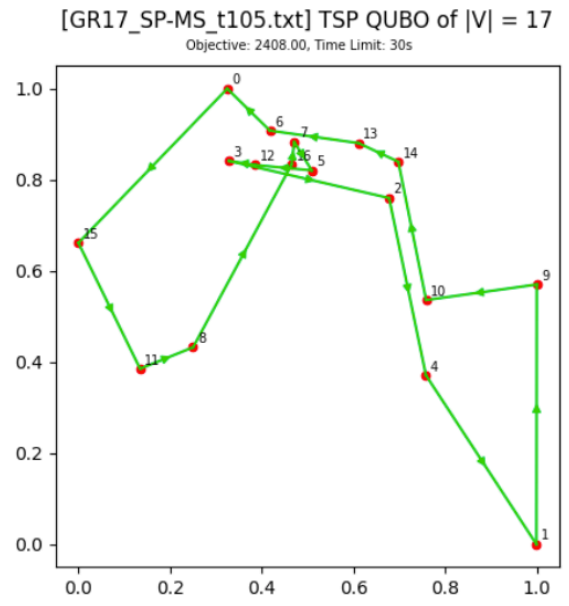


Figure 5.6 Experimental results of spanner graph theory with the highest time limit in our experiment. We don't want the time limit to be too high because we have limited access, and we want to measure the performance of our algorithm in stressful situations, considering the time it takes businesses to wait to solve the problems.

5.3 Conclusion of Approach 3: Supervised Deep Neural Network for a TSP Graph Problem

The two methods mentioned earlier, Matrix Sparsification and Graph Spanner, have provided promising results in terms of cost reduction when running the QUBO-based TSP problem on the D'Wave Leap Hybrid Solver quantum computer. However, we found that these results did not meet our criteria for project closure. Our ultimate goal is to obtain results that are theoretically sound and practically applicable in industrial settings, aligning with the objectives of our research project. Therefore, we must explore new methodologies to address this challenge effectively. We are committed to finding innovative approaches that can provide solutions capable of real-world implementation within the industrial sector.

The challenge we currently face revolves around simplifying the problem to expedite QUBO computations within limited timeframes, akin to what the industrial sector is encountering. Given the vast array of data and problem formats, we recognize that while certain theories may work in specific scenarios, they may not offer comprehensive solutions. Thus, our focus is on identifying patterns for this graph problem. We have turned our attention to machine learning, particularly as computer engineering students, and we are comfortable incorporating AI to tackle our challenges. Fortunately, we are not starting from scratch, as we already possess knowledge from our previous methods. Hence, in designing the neural network architecture, we can leverage practical knowledge or experimentally proven features to feed into our model. Our model is not specifically tailored to solve the TSP problem; instead, we aim to create a model that aids in sparsifying the TSP problem formulated into QUBO for quantum computation. Therefore, our model does not necessarily need to be state-of-the-art, as we can access quantum computers or quantum-inspired hardware provided by D'Wave to solve this problem.

Upon examining the results of our experimentation, as depicted in Figure 5.7, we were pleasantly surprised to observe a marked improvement in the outcomes. The quantum computations or solutions obtained from D'Wave provided the best solutions within the minimal time limit we set for this problem statement.

This underscores our capacity to effectively tackle QUBO problems about combinatorial optimization, such as the Traveling Salesperson Problem (TSP), and demonstrates the potential for practical implementation in real-world industrial settings. As illustrated in Figure 5.8, this achievement highlights the promising prospects of leveraging quantum computing and machine learning methodologies to address complex optimization challenges, paving the way for transformative applications across various industries.

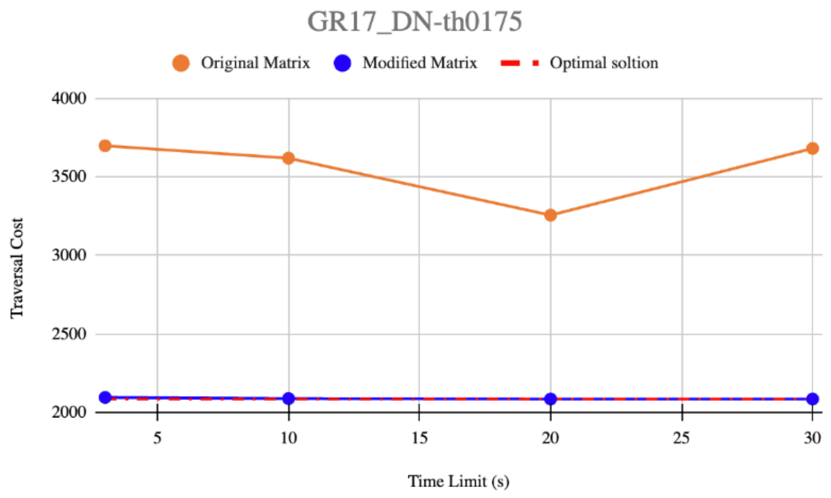


Figure 5.7 The application of our graph neural networks has yielded results that approach optimal or near-optimal solutions in some instances of experimentation, even within a time limit as short as 3 seconds. This outcome underscores the effectiveness of our approach in rapidly generating high-quality solutions to complex optimization problems. By harnessing the power of graph neural networks, we've unlocked solutions that previously seemed out of reach within such constrained time frames. This capability holds significant promise for enhancing decision-making processes and driving efficiency across various applications, showcasing the transformative potential of our experiment.

In conclusion, our experimentation journey has been marked by challenges and breakthroughs, all aimed at addressing the fundamental objective of our research. Through methods such as matrix sparsification and graph neural networks, we have achieved promising results in simplifying the TSP problem for quantum computing environments. While our current approaches have shown effectiveness in reducing computational costs and improving solution quality, they also underscore the need for further exploration and innovation. Looking ahead, we remain committed to exploring novel techniques, potentially leveraging insights from machine learning and graph theory, to devise robust solutions capable of addressing real-world TSP instances effectively. By continuing to refine our methodologies and collaborating across disciplines, we strive to contribute meaningfully to the advancement of quantum-inspired optimization techniques and their practical applications in various industries.

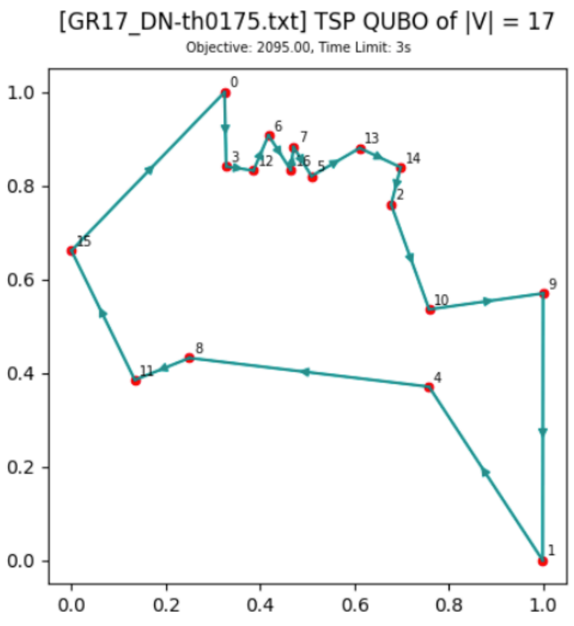


Figure 5.8 The traversal of paths in the GR17 problem yields results that are near the optimal value of 2095 compared to 2085 and is of significant importance in illustrating the efficacy of the methods and models employed in solving this problem. Occasionally, we may obtain directly optimal solutions. However, due to the probabilistic nature of quantum computers, each run may yield varying results.

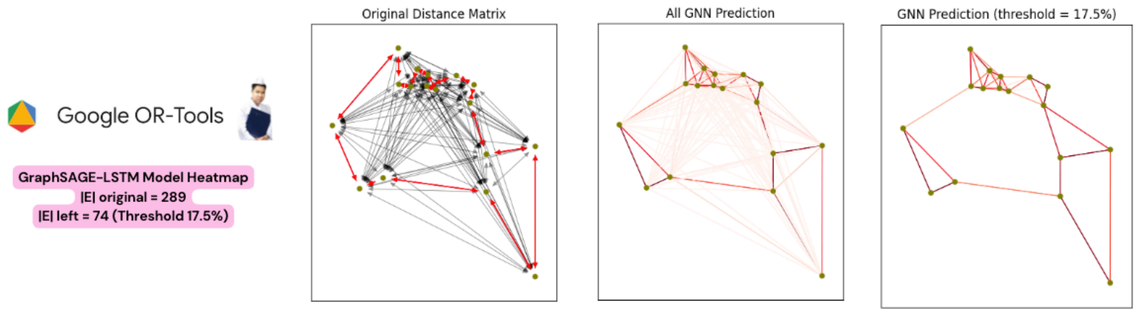

Google OR-Tools
GraphSAGE-LSTM Model Heatmap
|E| original = 289
|E| left = 74 (Threshold 17.5%)
Original Distance Matrix
All GNN Prediction
GNN Prediction (threshold = 17.5%)

Figure 5.9 The GR17 problem as input data to our model, which had been trained using data from Google OR-Tools, known for providing reasonable solutions quickly. Subsequently, we visualized this input data as a heatmap, representing the probability of each edge being part of the best solution graph. This visualization guides the QUBO solver, indicating that it should prioritize the edges identified by our machine-learning model. We have confidence that edges not determined by the ML model will likely not be part of our TSP solution.

REFERENCES

1. Mark Lewis and Fred Glover, 2017, “Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis,” **Networks**, vol. 70, no. 2, pp. 79–97, 2017.
2. AI Pakhomchik, S Yudin, MR Perelshtein, A Alekseyenko, and S Yarkoni, 2022, “Solving workflow scheduling problems with QUBO modeling,” **arXiv preprint arXiv:2205.04844**, 2022.
3. Fred Glover, Gary Kochenberger, Rick Hennig, and Yu Du, 2022, “Quantum bridge analytics I: a tutorial on formulating and using QUBO models,” **Annals of Operations Research**, vol. 314, no. 1, pp. 141–183, 2022.
4. Fred Glover, Gary Kochenberger, Moses Ma, and Yu Du, 2022, “Quantum Bridge Analytics II: QUBO-Plus, network optimization and combinatorial chaining for asset exchange,” **Annals of Operations Research**, vol. 314, no. 1, pp. 185–212, 2022.
5. Ehsan Zahedinejad and Arman Zaribafiyan, 2017, “Combinatorial optimization on gate model quantum computers: A survey,” **arXiv preprint arXiv:1708.05294**, 2017.
6. Fred Glover, Mark Lewis, and Gary Kochenberger, 2018, “Logical and inequality implications for reducing the size and difficulty of quadratic unconstrained binary optimization problems,” **European Journal of Operational Research**, vol. 265, no. 3, pp. 829–842, 2018.
7. Wai Shing Fung, Ramesh Hariharan, Nicholas JA Harvey, and Debmalya Panigrahi, 2011, “A general framework for graph sparsification,” in **Proceedings of the forty-third annual ACM symposium on Theory of computing**, 2011, pp. 71–80.
8. Juan Francisco Ariño Sales and Raúl Andres Palacios Araos, 2023, “Adiabatic Quantum Computing for Logistic Transport Optimization,” **arXiv preprint arXiv:2301.07691**, 2023.
9. William Evans and Lucca Morais de Arruda Siaudzionis, 2023, “On path-greedy geometric spanners,” **Computational Geometry**, vol. 110, pp. 101948, 2023.
10. Yong Shi and Yuanying Zhang, 2022, “The neural network methods for solving Traveling Salesman Problem,” **Procedia Computer Science**, vol. 199, pp. 681–686, 2022.
11. Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song, 2017, “Learning combinatorial optimization algorithms over graphs,” **Advances in neural information processing systems**, vol. 30, 2017.
12. Marcelo Prates, Pedro HC Avelar, Henrique Lemos, Luis C Lamb, and Moshe Y Vardi, 2019, “Learning to solve np-complete problems: A graph neural network for decision tsp,” in **Proceedings of the AAAI Conference on Artificial Intelligence**, 2019, vol. 33, pp. 4731–4738.
13. Yimeng Min, Yiwei Bai, and Carla P Gomes, 2023, “Unsupervised Learning for Solving the Travelling Salesman Problem,” **arXiv preprint arXiv:2303.10538**, 2023.
14. Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun, 2020, “Graph neural networks: A review of methods and applications,” **AI open**, vol. 1, pp. 57–81, 2020.
15. Thomas N Kipf and Max Welling, 2016, “Semi-supervised classification with graph convolutional networks,” **arXiv preprint arXiv:1609.02907**, 2016.
16. Chaitanya K Joshi, Thomas Laurent, and Xavier Bresson, 2019, “An efficient graph convolutional network technique for the travelling salesman problem,” **arXiv preprint arXiv:1906.01227**, 2019.
17. Gerhard Reinelt, 1991, “TSPLIB—A traveling salesman problem library,” **ORSA journal on computing**, vol. 3, no. 4, pp. 376–384, 1991.