POLITECNICO DI TORINO

Master degree course in Automation and Intelligent Cyber-Physical Systems

Master Degree Thesis

Space Debris Removal Optimization Using Quantum Annealing

Supervisors
Prof. Carlo Novara

Correlatore:
PhD. Mattia Boggio
Eng. Deborah Volpe

Candidate
Michele Gagliardi

April 2025

Enjoy the ***SOS***

Summary

The exponential growth of space debris in Low Earth Orbit (LEO) poses a significant challenge to the sustainability of space operations. Despite preventive measures aimed at limiting debris generation, they remain insufficient to address the increasing accumulation of defunct satellites, rocket stages, and collision fragments. Active Debris Removal (ADR) has emerged as a promising solution, particularly in multi-target missions, which require solving complex combinatorial optimization problems similar to the Traveling Salesman Problem (TSP) to maximize the efficiency of the missions, minimizing fuel use and mission duration. This thesis explores the application of Quantum Annealing (QA) and Hybrid Quantum Annealing (HQA) to optimize multi-target ADR missions. Specifically, it introduces a Quadratic Unconstrained Binary Optimization (QUBO) model tailored for ADR using quantum computing frameworks to enhance solution efficiency. The research develops a generalized quadratization method to reduce computational complexity, enabling large-scale mission planning. Additionally, it proposes a novel constraint-handling strategy, embedding mission constraints in post-processing to improve quantum solver performance. The proposed approach is applied to real-world satellite debris datasets and benchmarked against classical metaheuristic optimizers, including Simulated Annealing (SA), Tabu Search (TS), and Genetic Algorithms (GA). The results highlight the potential of quantum optimization for ADR mission planning, offering a scalable and computationally efficient solution. This research represents one of the first applications of quantum computing to orbital debris management, contributing to the advancement of sustainable space operations.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my family, whose unwavering support, encouragement, and patience have been invaluable throughout my academic journey. Their belief in me has always provided strength and motivation.

I extend my sincere appreciation to my supervisor, Prof. Carlo Novara, and to my co-supervisors, PhD Mattia Boggio and Eng. Deborah Volpe, for their continuous guidance, insightful advice, invaluable suggestions, and consistent support during this research.

Finally, heartfelt thanks go to all my friends, both those I've known my entire life and those I've had the pleasure to meet during my university years, including the remarkable colleagues from Team Diana. Your friendship, shared experiences, and constant encouragement have made this journey truly memorable.

Thank you all.

Contents

I Introduction and Background 1

1 Background and Motivation 3
1.1 Historical Evolution of Space Debris 3
1.2 Current State of the Orbital Environment 4
1.3 Active Debris Removal (ADR) and Its Importance 6

2 Problem Statement and Research Motivation 9
2.1 The Active Debris Removal (ADR) Optimization Challenge 9
2.2 Quantum Annealing for ADR Optimization 10
2.2.1 Quantum Computing in Aerospace Applications 10
2.3 Quantum Annealing for ADR Optimization 10
2.4 Objectives and Contributions 11
2.5 Thesis Structure Overview 12

II Theoretical Foundations and Modeling Tools 15

3 NP Problems and Combinatorial Optimization 17
3.1 Introduction to NP Problems 17
3.2 Optimization Paradigms and Mathematical Modeling 19
3.2.1 Linear vs. Nonlinear Optimization 19
3.2.2 Combinatorial Optimization 20
3.2.3 Binary Optimization and Constraint Handling 20
3.2.4 From Linear Models to QUBO/HUBO 21

4 QUBO and HUBO Mathematical Formulations 23
4.1 Quadratic Unconstrained Binary Optimization (QUBO) 23
4.1.1 Why QUBO Matters 24
4.2 Higher-Order Unconstrained Binary Optimization (HUBO) 24
4.3 From HUBO to QUBO – Problem Mapping 25

5 Quadratization Techniques 27
5.1 Classical Quadratization Methods 27
5.1.1 Quadratization of a Cubic Term: A Concrete Example . . . 28
5.1.2 Rosenberg's Quadratization Method (1975) 28
5.1.3 Quadratization of Quartic and Higher-Order Terms 29
5.2 Efficiency Considerations in Quadratization 29

6 Solvers and Optimization Algorithms 31
6.1 Heuristic and Metaheuristic Approaches 31
6.2 Quantum Annealing (QA) and Hybrid Quantum Annealing (HQA) 32
6.2.1 How Quantum Annealers Work 33
6.2.2 Solving a QUBO Problem with a Quantum Annealer 33
6.2.3 The Adiabatic Condition and Annealing Time Constraints . 35
6.2.4 QUBO to Ising Model Mapping 35
6.2.5 Hybrid Quantum Annealing (HQA) 36
6.3 Classical vs. Quantum Optimization Methods: A Comparative Discussion 36
6.3.1 Strengths of Classical Methods 36
6.3.2 Limitations of Classical Methods 37
6.3.3 Strengths of Quantum Optimization 37
6.3.4 Limitations of Quantum Optimization 38
6.3.5 Potential Impact and Outlook 39

III Problem Modeling and Quadratization Strategies 41

7 Problem Description and Mathematical Formalization 43
7.1 The Space Debris Collection Problem 43
7.2 Definition of the Capture Pattern and Variables 44
7.3 TOF 45
7.3.1 RAAN Alignment Time t_R 46
7.3.2 Hohmann Transfer Time t_H 47
7.3.3 Rendezvous Waiting Time: t_W 47
7.3.4 Capture Time t_{cat} 49
7.3.5 Release Time t_{ril} 49
7.4 Delta-v Δv 50
7.5 Desirability 51
7.6 Mathematical Formulation of Objectives and Constraints 51
7.6.1 Variables and Parameters 51
7.6.2 Objective Function 52
7.6.3 Model Constraints 53
7.6.4 Final Considerations 53

8 Complexity Analysis and Theoretical Challenges 55
8.1 Analysis of the Polynomial Objective Function and Its Complexity . 55
8.1.1 Analysis of the TOF Term 56
8.1.2 Analysis of the Linear Terms 56
8.2 Impact of Increasing Debris and Capture Events on Computational Effort 56
8.2.1 Effect of the Number of Debris (D) 57
8.2.2 Effect of the Number of Capture Events (C) 57
8.3 Overall Considerations 57

9 Quadratization Strategies and Innovations 59
9.1 Traditional Quadratization of Polynomial Problems (Case >2 Captures) 59
9.2 Novel Generalized Quadratization Method 60
9.2.1 Context and TOF Problematic 60
9.2.2 Approximation via the Synodic Period 61
9.2.3 Advantages of the New Methodology 62
9.3 Comparative Analysis: Performance and Computational Efficiency . 62

10 Conversion to QUBO/HUBO Format 63
10.1 Mapping Techniques to QUBO/HUBO Format 63
10.2 Variable Structure and Costas Array Encoding 64
10.3 Definition of Penalty Terms and Variable Reduction Strategies . . . 65
10.3.1 Decision of Exclusive Use of P_{matrix} in QA 66
10.3.2 Why a High Lambda Coefficient Worsens Annealing 66
10.3.3 The Role of Chain Strength and Embedding Management . 67
10.4 Comparison Between QA and HQA/SA Strategies 67

IV Implementation, Experimentation, and Comparative Analysis **69**

11 Quantum Annealing Implementation 71
11.1 Hardware Description 71
11.1.1 Quantum Annealer Architecture 71
11.1.2 Hardware Specifications and Performance 73
11.2 Development Environment 74
11.2.1 Software Tools and Frameworks 74
11.2.2 Programming and Workflow Integration 75
11.3 Dataset Handling 76
11.3.1 Source of Satellite Debris Data 76
11.3.2 Data Filtering and Preprocessing 76
11.3.3 QUBO Formulation from Debris Data 77

11.4 Overall Workflow 77

12 Comparative Analysis of Quantum and Classical Approaches 79
12.1 Quantum Annealing Constraint Management 79
12.2 Solver Comparison and Scalability 81
12.3 Advantage of the new Quadratization Method 83
12.4 Discussion 87

13 MATLAB/Simulink Mission Simulation 91
13.1 Simulation Objectives 91
13.2 Simulation Environment 92
13.3 Simulation Workflow 92
13.4 Results and Verification 93
13.5 Discussion and Implications 94

V Conclusions 97

14 Conclusions and Future Directions 99

Part I

Introduction and Background

Chapter 1

Background and Motivation

1.1 Historical Evolution of Space Debris

The Space Age began in 1957 with the launch of Sputnik 1, marking not only humanity's first artificial satellite but also the creation of the first human-made orbital debris—namely, the spent rocket stage and the inert satellite itself. During the 1960s, the increasing frequency of space launches, along with occasional in-orbit explosions and tests, contributed to a steadily growing debris population.

A pivotal event occurred in 1961 when the Thor-Ablestar upper stage became the first known satellite to undergo fragmentation in orbit, producing over 200 trackable fragments. However, the implications of space debris as a long-term hazard became more apparent in 1978, when NASA scientist Donald Kessler introduced a theoretical model predicting that, beyond a critical threshold, debris collisions could trigger a self-sustaining cascade of fragmentations—an effect later known as the Kessler Syndrome [1]. This hypothesis raised concerns about the long-term usability of Low Earth Orbit (LEO) and the necessity of mitigation strategies [2]. Although early fragmentation events occurred, the full impact of this phenomenon had yet to be observed.

Throughout the late 20th century, each decade witnessed an increasing number of launches and occasional major fragmentation events, leading to a steady rise in the number of tracked objects in orbit. However, two incidents in the late 2000s significantly exacerbated the space debris problem:

- 2007: China conducted an anti-satellite (ASAT) test, deliberately destroying the defunct Fengyun-1C weather satellite. This event alone generated over 2,000 trackable debris fragments, making it one of the most significant single sources of orbital debris [3].

- 2009: An accidental collision occurred between the active Iridium 33 satellite and the defunct Cosmos-2251, producing more than 1,800 additional fragments [4].

These incidents resulted in an immediate and dramatic increase in the debris population within LEO, intensifying global concerns over space traffic management and long-term orbital sustainability. Figure 1.1 illustrates the exponential growth in debris count over time, correlating with the growing launch rates and cumulative effects of accidental breakups and deliberate satellite destructions.

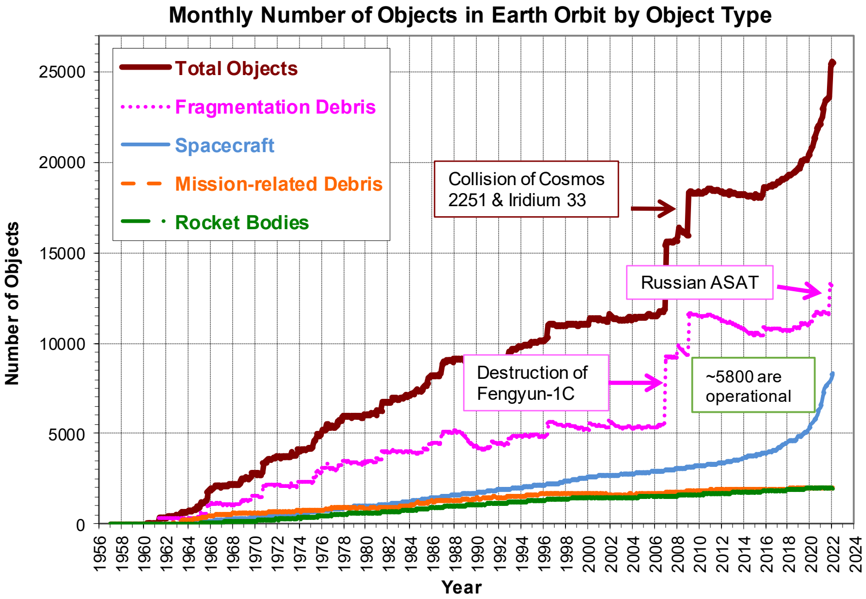


Figure 1.1: Orbital Debris Population Trend. This graph shows the number of objects in orbit—from small fragments to objects roughly the size of a baseball—tracked by the NASA Orbital Debris Program Office. It highlights significant increases following key events (e.g., the Iridium 33–Cosmos 2251 collision and the Chinese ASAT test), which have dramatically raised the total number of debris objects. (Source: *Space Debris and ISS*, NASA, DAA 20230010751) [5]

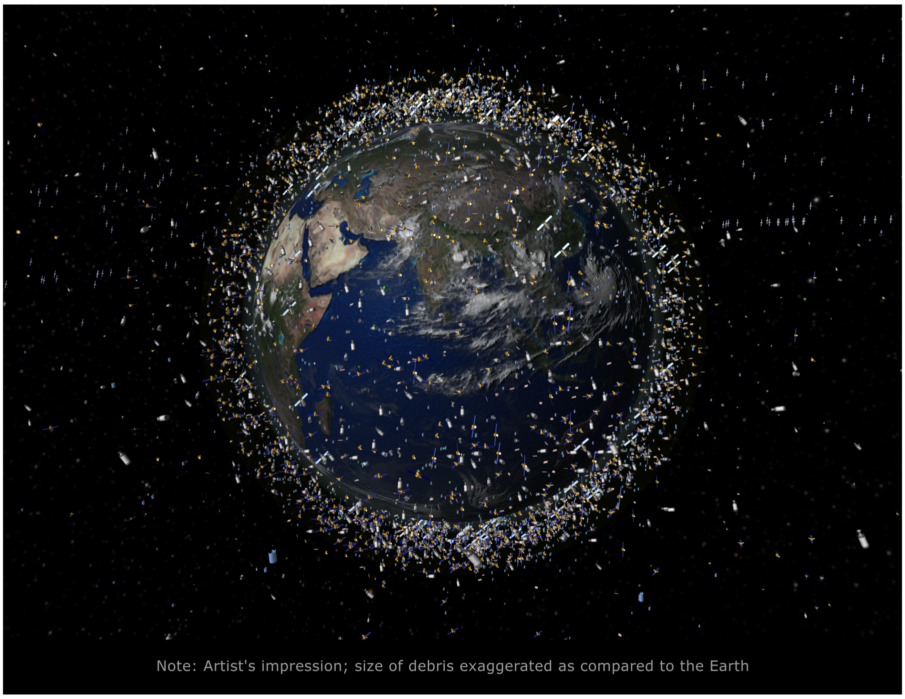
1.2 Current State of the Orbital Environment

The near-Earth space environment is increasingly congested with debris of various sizes, posing significant challenges to satellite operations and future space missions. According to the European Space Agency (ESA), as of 2019, the estimated debris population included:

- 40,500 objects larger than 10 cm
- 1,100,000 objects between 1–10 cm
- 130 million fragments measuring 1 mm to 1 cm [6]

Of the total orbital population, only a few thousand objects are active satellites, while the vast majority consist of defunct payloads, spent rocket stages, and fragmentation debris. These objects travel at velocities exceeding 7 km/s, meaning that even millimeter-sized paint flakes have the potential to damage or disable a spacecraft upon impact. A collision with a 10 cm debris fragment could be catastrophic, completely destroying a satellite and generating thousands of additional fragments. A notable example of such risks occurred in 2021 when a small debris strike punctured the International Space Station's (ISS) robotic arm, underscoring the persistent threat posed even by sub-centimeter fragments [7].

The spatial distribution of debris is heaviest in Low Earth Orbit (LEO), particularly below 2,000 km altitude, where approximately 70% of all cataloged objects are concentrated. Figure 1.2 provides a visual representation of the density of tracked debris around Earth, with object sizes exaggerated for clarity.



Note: Artist's impression; size of debris exaggerated as compared to the Earth

Figure 1.2: Representation of debris objects in low-Earth orbit. Crowded clusters of dots (not to scale) illustrate the concentration of man-made objects around Earth, especially in LEO. Over 70% of catalogued orbital objects reside within 2000 km of Earth's surface. The ring of objects along the equator depicts the geostationary belt. (Image credit: ESA)

The operational consequences of this debris environment are substantial. Satellite operators now routinely perform collision avoidance maneuvers to mitigate impact risks. For instance, an average satellite in LEO may receive dozens of close-approach alerts per week, and if the estimated collision probability exceeds 1 in

10,000, operators must adjust the satellite's trajectory, consuming valuable fuel and operational resources. The ISS alone has had to conduct multiple orbit adjustments per year to avoid potential collisions. In one critical event, in October 2022, a fragment from the 2021 Russian ASAT test of Cosmos-1408 necessitated an urgent maneuver to increase the ISS altitude and avoid a predicted impact trajectory [8].

An even more pressing concern is the potential onset of Kessler Syndrome. While Kessler's model was originally a theoretical concern, recent decades have shown increasing evidence of its potential onset. ESA data indicates a 50% rise in key orbital debris levels over the last five years [9], driven by major fragmentation events such as the 2009 Iridium-Cosmos collision. The growing frequency of uncontrolled breakups suggests that, without decisive intervention, the combination of ongoing launches (particularly large constellations) and spontaneous collisions could accelerate this process, jeopardizing the long-term sustainability of space operations.

To address this escalating issue, international space agencies and organizations have introduced guidelines aimed at limiting further debris generation. One of the most significant frameworks is the Space Debris Mitigation Guidelines, published in 2002 by the Inter-Agency Space Debris Coordination Committee (IADC) [10].

1.3 Active Debris Removal (ADR) and Its Importance

The escalating risks associated with space debris proliferation necessitate not only mitigation strategies but also active intervention to remove existing debris from orbit. While traditional debris mitigation efforts—such as collision avoidance maneuvers and post-mission disposal guidelines—have slowed debris accumulation, they are insufficient to reverse the current trend. Without additional measures, the number of hazardous objects in orbit will continue to rise, increasing the likelihood of collisional cascading and exacerbating long-term sustainability concerns.

Active Debris Removal (ADR) refers to the deliberate capture and removal of debris objects using specialized spacecraft. Unlike mitigation strategies that aim to prevent new debris formation, ADR seeks to reduce the existing debris load by removing the most problematic objects before they fragment further or cause additional collisions.

The rationale for ADR is supported by numerical simulations showing that even a modest removal rate—as low as five large debris objects per year—could significantly stabilize the LEO environment and slow the progression toward collisional cascading [11, 12, 13]. Targeting large, intact rocket bodies and defunct satellites is particularly important, as these objects represent the highest potential for future fragmentation due to fuel leaks, structural degradation, and accidental impacts.

Several pioneering ADR missions are in development to demonstrate the feasibility of debris removal. For example, ESA has contracted a commercial start-up, ClearSpace SA, to conduct the ClearSpace-1 mission in 2025 — which aims to rendezvous with a leftover Vega rocket adapter (about 100 kg in mass) and deorbit it safely [14]. This will be the world's first attempt to actively capture and dispose of an uncontrolled object in orbit.

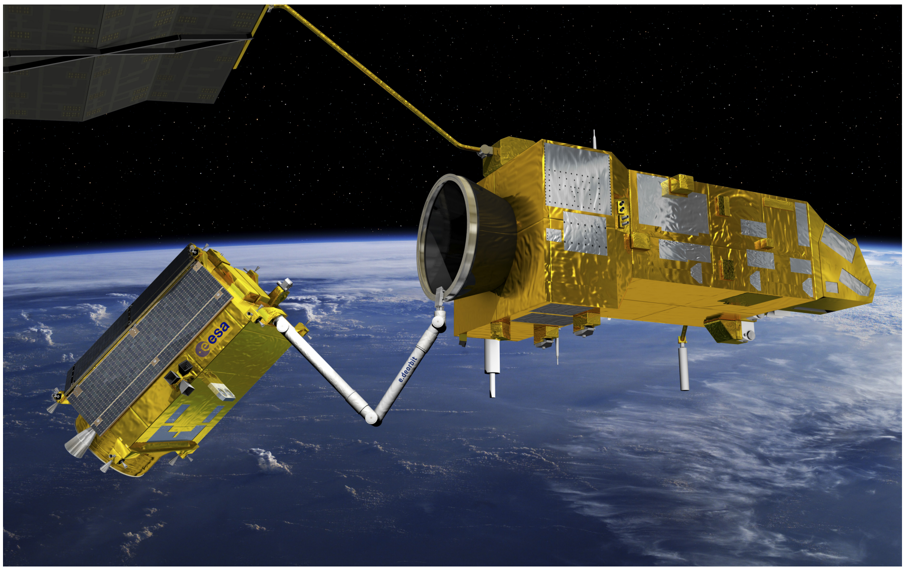


Figure 1.3: Artist's illustration of an active debris removal concept: ESA's proposed e.Deorbit mission using a robotic arm to capture the defunct Envisat satellite ($\approx$ 8.5 tonnes) in low Earth orbit. After grappling the target, the combined stack would be deorbited for controlled atmospheric re-entry. e.Deorbit was planned as the first ADR mission (originally slated for mid-2020s), intended to demonstrate safe capture and disposal of a large uncooperative object. (Image credit: ESA/D. Ducros)

Despite the clear need, scaling up ADR presents economic challenges. Each removal mission is costly; for instance, the ClearSpace-1 contract is valued at €86 million for a single object removal. At this price, removing dozens of objects annually would require substantial funding. However, the cost of inaction could be far higher — an uncontrolled growth of debris could lead to frequent satellite losses, interruptions to services, and lost access to entire orbital regions. Recognizing this, agencies are investing in ADR technologies and partnerships to drive down costs and improve efficiency. Initiatives like ESA's Clean Space and NASA's Space Technology programs are fostering innovations (e.g. low-cost capture mechanisms, better tracking, and AI for autonomous rendezvous) that will feed into future ADR efforts. The ultimate goal is to make debris removal a routine part of space operations. Achieving that will likely require not only technical breakthroughs but also optimization of ADR mission plans – which is where advanced computational approaches, including emerging quantum computing techniques, come into play.

Chapter 2

Problem Statement and Research Motivation

2.1 The Active Debris Removal (ADR) Optimization Challenge

Considering the issues previously discussed, this thesis focuses on optimizing multi-target Active Debris Removal (ADR) missions. In these missions, a single chaser spacecraft is tasked with capturing and disposing of multiple debris objects within one mission. The challenge lies in determining the optimal sequence and trajectory that minimizes both the total time of flight (TOF) and fuel consumption (Δv), while satisfying complex operational constraints such as orbital dynamics (e.g., RAAN alignment, Hohmann transfer maneuvers) and target prioritization based on debris desirability.

Due to its combinatorial nature—resembling the well-known Traveling Salesman Problem (TSP) [15]—the ADR mission planning problem is classified as NP-hard. The number of possible capture sequences grows exponentially with the number of debris targets, making exhaustive search methods computationally intractable. The problem is further complicated by the time-dependent dynamics of orbital mechanics, as debris objects follow distinct trajectories and precession rates. Given the impracticability of obtaining exact solutions for real-world scenarios, various heuristic and metaheuristic approaches have been proposed to compute near-optimal solutions within feasible time constraints.

Notable heuristic techniques include tree search procedures such as Beam Search [16], Ant Colony Optimization [17], and A* search [18], which iteratively build solutions by sequentially adding debris targets. Metaheuristic approaches such as Simulated Annealing (SA) [19] and Genetic Algorithms (GAs) [20] refine a set of candidate solutions through iterative improvement. More recently, Reinforcement Learning (RL) and Machine Learning (ML) techniques [21, 22] have been explored, offering a data-driven approach to optimizing debris removal sequences.

However, due to the inherent complexity of the ADR optimization problem, even state-of-the-art metaheuristic methods may struggle to identify high-quality solutions or may require prohibitive computational resources as the problem size increases. These limitations have motivated the exploration of alternative computational paradigms, including the emerging field of quantum computing.

2.2 Quantum Annealing for ADR Optimization

2.2.1 Quantum Computing in Aerospace Applications

Quantum computing has gained increasing attention in aerospace applications, particularly in mission scheduling and optimization problems. In 2016, NASA outlined the opportunities and challenges of quantum annealing (QA) for space-related tasks, with a focus on path planning and scheduling [23]. Since then, significant progress has been made in this area, particularly in optimizing Earth Observation (EO) mission scheduling, as demonstrated by recent studies [24, 25].

Despite this growing interest, the application of quantum annealing to ADR remains largely unexplored, with only a few pioneering works addressing debris removal optimization. In [26], the ADR mission planning problem is tackled using a hybrid approach that combines Artificial Neural Networks (ANNs) with the Fujitsu Digital Annealer, a quantum-inspired digital optimization architecture [27]. Another study [28] applies quantum annealing to the problem of atmospheric disposal via the uncontrolled reentry of small debris. However, a comprehensive, quantum-based framework for large-scale multi-target ADR planning has yet to be developed.

2.3 Quantum Annealing for ADR Optimization

Quantum Annealing (QA) [29] is a promising approach for solving combinatorial optimization problems such as ADR mission planning. Unlike classical optimization methods, which explore solutions sequentially or heuristically, QA leverages quantum mechanics—specifically superposition and quantum tunneling—to explore vast solution spaces more efficiently. This ability makes it particularly well-suited for problems with a rugged cost landscape, where traditional algorithms often get trapped in local minima.

In this context, the ADR optimization problem can be formulated as a Quadratic Unconstrained Binary Optimization (QUBO) model. By encoding decision variables into a binary matrix representing the capture sequence of debris targets, the problem becomes compatible with quantum annealers. This approach enables the simultaneous evaluation of an exponential number of solutions, potentially yielding higher-quality mission plans within a shorter time frame [30, 31].

While quantum computing is still in its early stages, hybrid quantum-classical methods have already demonstrated competitive performance on NP-hard routing and scheduling problems, suggesting potential advantages for ADR optimization. The benefits of quantum annealing for ADR are compelling:

- **Fuel efficiency:** Optimized debris removal sequences can significantly reduce the Δv required.
- **Shorter mission durations:** More efficient trajectories reduce time of flight (TOF), allowing for faster debris clearance.
- **Scalability:** Quantum annealers can handle large-scale ADR problems more efficiently than classical heuristics.

Despite current hardware limitations—such as qubit count and noise—the combinatorial nature of ADR planning makes it an ideal candidate for quantum-enhanced optimization. Even incremental improvements in solution quality or computational efficiency could translate into substantial cost savings, making large-scale debris removal economically viable.

By integrating quantum annealing into ADR mission planning, this research explores new strategies to address the growing space debris crisis, improve sustainability in low-Earth orbit, and ensure the long-term viability of future space activities.

2.4 Objectives and Contributions

The primary objectives of this thesis are to:

1. **Formulate a Comprehensive ADR Optimization Model:** Develop a rigorous mathematical formulation that captures the critical parameters of ADR missions—including time of flight, delta-v, and debris desirability—while integrating complex orbital dynamics constraints such as RAAN alignment and Hohmann transfer maneuvers.
2. **Develop a QUBO Model for ADR Mission Planning:** Translate the ADR problem into a Quadratic Unconstrained Binary Optimization (QUBO) formulation suitable for quantum annealing. This includes the design of a novel quadratization method that approximates higher-order terms using the synodic period, thereby reducing the problem's computational complexity from exponential to polynomial order.

3. **Integrate Quantum Optimization Techniques:** Implement and benchmark the formulated QUBO model on quantum annealing hardware using both pure Quantum Annealing (QA) and Hybrid Quantum Annealing (HQA). The performance of these quantum-based approaches is compared with classical metaheuristic methods such as Simulated Annealing, Tabu Search, and Genetic Algorithms.

4. **Validate with Real-World Data:** Test the developed framework using real satellite debris datasets derived from Two-Line Element (TLE) files. This validation demonstrates the practical applicability and potential advantages of quantum-enhanced optimization in real-world ADR mission-planning scenarios.

5. **Propose a Novel Constraint-Handling Strategy:** Address the challenges posed by the combinatorial explosion of potential debris capture sequences by introducing a post-processing constraint management strategy that significantly improves solver performance without compromising solution quality.

In summary, this thesis contributes a novel, quantum-optimized framework for multi-target ADR mission planning. By bridging advanced quantum computing techniques with the critical challenges of space debris removal, the research not only advances the state-of-the-art in orbital debris mitigation but also lays the groundwork for more sustainable space operations in the future.

2.5 Thesis Structure Overview

This thesis is organized into five main parts, each addressing a specific aspect of the Active Debris Removal (ADR) optimization problem via Quantum Annealing (QA).

- **Part I: Introduction and Background** sets the context and rationale for this research. Chapter 1 reviews the historical development and current challenges of orbital debris, highlighting the critical role of ADR in ensuring the long-term sustainability of space activities. Chapter 2 defines the ADR optimization problem, outlines the research objectives and contributions, and presents the structure of the thesis.

- **Part II: Theoretical Foundations and Modeling Tools** provides the conceptual framework underpinning the study. Chapter 3 introduces NP-hard problems and combinatorial optimization techniques relevant to ADR mission planning. Chapters 4 and 5 explore the QUBO and HUBO formulations and quadratization methods required for quantum optimization. Chapter 6 compares classical and quantum optimization algorithms, offering a foundation for subsequent experimental analysis.

- **Part III: Problem Modeling and Quadratization Strategies** formalizes the ADR optimization problem. Chapters 7 and 8 detail the mathematical modeling and complexity analysis. Chapter 9 presents an innovative quadratization approach to enhance computational efficiency, while Chapter 10 focuses on translating the ADR problem into QUBO/HUBO formats suitable for quantum solvers.

- **Part IV: Implementation, Experimentation, and Comparative Analysis** describes the implementation of the proposed optimization framework. Chapters 11 and 12 present the experimental setup, performance evaluation, and comparative analysis between quantum and classical solvers. Chapter 13 validates the theoretical models through MATLAB/Simulink-based simulations of ADR missions.

- **Part V: Conclusions** summarizes the research findings and discusses future directions, emphasizing the potential contributions of quantum computing to sustainable space operations.

Part II

Theoretical Foundations and Modeling Tools

Chapter 3

NP Problems and Combinatorial Optimization

3.1 Introduction to NP Problems

In computational complexity theory, problems are classified based on the computational resources required to solve them. A problem belongs to the class NP (Nondeterministic Polynomial time) if a proposed solution can be verified in polynomial time by a deterministic algorithm [32]. However, this does not imply that solutions can be efficiently computed—only that they can be efficiently verified. Many decision and optimization problems of practical significance belong to NP, where solution verification is polynomial-time, but solution discovery may be computationally intractable.

A problem is classified as NP-hard if it is at least as difficult as the hardest problems in NP [33]. Formally, a problem H is NP-hard if every problem in NP can be polynomial-time reduced to H. If, in addition, the problem itself is in NP (i.e., both NP-hard and verifiable in polynomial time), it is termed NP-complete. NP-complete problems are the most challenging within NP — if any NP-complete problem is solved in polynomial time, it would imply $P = NP$, a fundamental unsolved question in theoretical computer science. It is widely conjectured that $P \neq NP$, suggesting that no polynomial-time algorithms exist for NP-complete problems [33].

A well-known example is the Traveling Salesman Problem (TSP) [32]. Given a set of n cities and pairwise distances, the objective is to determine the shortest possible tour visiting each city exactly once before returning to the starting point. The number of possible tours grows as $(n - 1)!/2$, making an exhaustive search infeasible even for moderate values of n. TSP is NP-hard in its optimization form and NP-complete in its decision version. Similarly, the Active Debris Removal (ADR) problem in orbital mechanics, which optimizes satellite removal sequences,

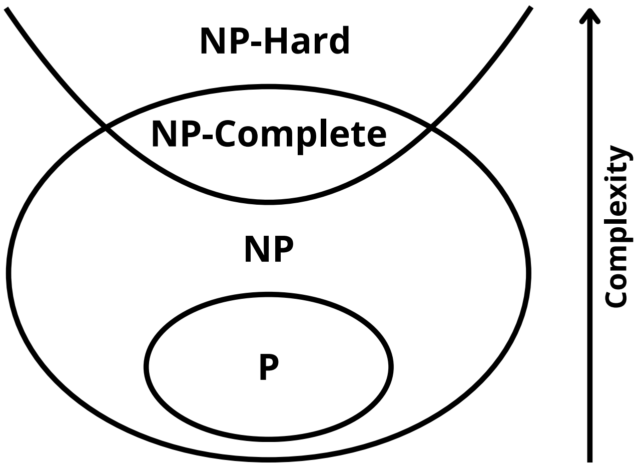
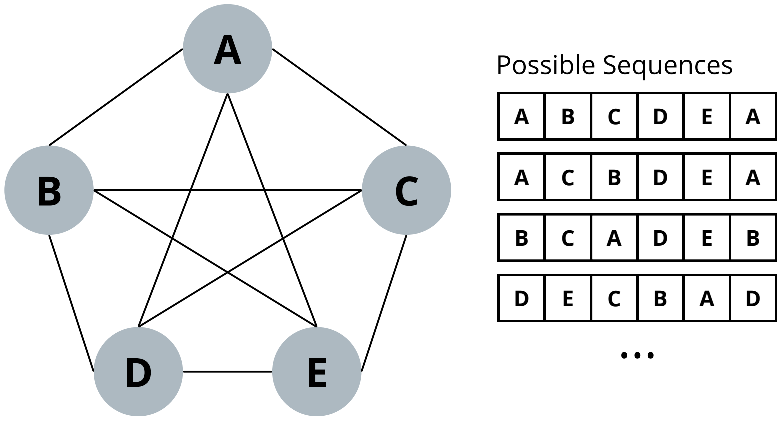


Figure 3.1: Diagram of Complexity Classes illustrating the relationships between P, NP, NP-Complete, and NP-Hard problems. This figure visually represents how these classes are nested and highlights the relative computational complexity of each.

shares structural similarities with TSP and is also NP-hard.



Possible Sequences

A	B	C	D	E	A
A	C	B	D	E	A
B	C	A	D	E	B
D	E	C	B	A	D

...

Figure 3.2: Example of an NP-Hard problem: the Traveling Salesman Problem (TSP). The diagram displays some of the several possible sequences for visiting cities, emphasizing the exponential increase in route combinations as the number of cities grows.

For NP-hard problems like TSP, exact algorithms (e.g., dynamic programming, integer programming) are impractical for large inputs due to exponential growth in computational complexity. Instead, researchers employ approximation algorithms and heuristic approaches to obtain near-optimal solutions in a reasonable time. Common heuristics and metaheuristics include Beam Search, Ant Colony Optimization, A* search, Simulated Annealing, and Genetic Algorithms, which leverage problem-specific properties and randomness to navigate the solution space efficiently. However, even sophisticated heuristics can struggle with large-scale instances, requiring extensive computational resources and fine-tuning while offering no formal guarantee of optimality.

In conclusion, the complexity classes NP, NP-hard, and NP-complete play a crucial role in understanding computational feasibility. Many real-world combinatorial optimization problems, such as scheduling, routing, and resource allocation, fall within these categories. Given the intractability of exact solutions, continued research explores improved heuristics, approximation techniques, and alternative computational paradigms, including quantum computing, to address NP-hard optimization problems effectively.

3.2 Optimization Paradigms and Mathematical Modeling

In mathematical optimization, problems are classified based on the nature of their objective functions and constraints. Two fundamental categories are continuous optimization, which includes linear and nonlinear programming, and combinatorial optimization, which involves discrete decision variables, often modeled using binary or integer values. This discussion focuses on combinatorial optimization, which aims to find an optimal solution over a discrete but potentially exponentially large search space.

3.2.1 Linear vs. Nonlinear Optimization

Linear Programming (LP) involves optimization problems where the objective function and constraints are linear functions of the decision variables. LP problems can be solved efficiently using algorithms such as the Simplex method or interior-point methods [34]. However, if we introduce integer or binary constraints on decision variables, the problem transforms into Integer Programming (IP) or Mixed-Integer Programming (MIP), which are generally NP-hard and significantly more difficult to solve exactly [35].

Nonlinear optimization arises when the objective function or constraints involve nonlinear terms. If the problem is convex, efficient algorithms exist, and any local optimum is also a global optimum [36]. However, non-convex problems present greater computational challenges as they may contain multiple local optima, making global optimization intractable.

3.2.2 Combinatorial Optimization

Combinatorial optimization involves problems where the solution space is discrete, often structured as permutations, subsets, or assignments. Classical examples include:

- Scheduling problems (e.g., job shop scheduling, course timetabling)
- Routing problems (e.g., Traveling Salesman Problem, Vehicle Routing Problem)
- Assignment problems (e.g., matching workers to tasks, facility location problems)
- Graph-based problems (e.g., network design, clique finding, coloring problems)

These problems are commonly modeled using binary (0-1) decision variables, where each variable represents a discrete choice—such as whether to include an item in a knapsack or whether an edge in a graph is used.

A powerful way to model such problems is through 0-1 Integer Linear Programming (0-1 ILP). For example, in the Traveling Salesman Problem (TSP), the binary variable x_{ij} may indicate whether the salesman travels directly from city i to city j. The objective function minimizes total travel distance, while constraints ensure that each city is visited exactly once and that subtours (cycles within the tour) are eliminated. However, due to the binary nature of the decision variables, such ILP models are typically NP-hard, requiring advanced solving techniques.

3.2.3 Binary Optimization and Constraint Handling

A particularly significant subclass of combinatorial optimization problems is binary optimization, where all decision variables are binary. These models are widely used in operations research, artificial intelligence, and theoretical computer science, including applications in SAT solvers, ILP solvers, and quantum optimization frameworks. The significance of binary models lies in their flexibility—many complex problems can be reformulated as binary problems through appropriate constraints and objective function transformations. Some modern solvers and even quantum annealers are designed to handle binary decision variables natively. By formulating a problem in terms of binary variables, we make it amenable to these solvers.

Traditional constrained optimization problems (such as ILPs) impose explicit feasibility conditions that solutions must satisfy. However, some solving approaches — such as quantum annealing — prefer unconstrained formulations. This has led to a modeling shift where hard constraints are incorporated into the objective function via penalty terms, effectively transforming constrained problems into unconstrained ones.

3.2.4 From Linear Models to QUBO/HUBO

Linear models, while widely used, may not always capture complex relationships between decision variables. In many practical applications, quadratic or higher-order interactions naturally arise. For instance, in scheduling, a quadratic term $x_i \cdot x_j$ might model a conflict cost if tasks i and j are scheduled together.

This leads to Quadratic Unconstrained Binary Optimization (QUBO), a formulation where the objective function consists of quadratic terms over binary variables. QUBO models are particularly important because they are directly solvable by modern quantum annealers [37] and specialized combinatorial optimization solvers.

If the objective function contains higher-order interactions (e.g., cubic, quartic terms), the problem is classified as Higher-Order Binary Optimization (HUBO). While these models can be more compact and expressive, they often require transformation into an equivalent quadratic form (a process known as quadratization) to leverage existing QUBO solvers.

Chapter 4

QUBO and HUBO Mathematical Formulations

As optimization models evolve to incorporate quadratic and higher-order interactions over binary variables, the QUBO (Quadratic Unconstrained Binary Optimization) and HUBO (Higher-Order Unconstrained Binary Optimization) frameworks become central in combinatorial optimization [38]. These formulations capture complex decision structures and are particularly relevant for emerging computational paradigms, including quantum computing and specialized classical solvers.

4.1 Quadratic Unconstrained Binary Optimization (QUBO)

QUBO problem consists of an objective function defined over n binary decision variables $x = (x_1, x_2, ..., x_n)$, where each x_i takes values in {0,1}. The term "unconstrained" signifies that any hard constraints have been incorporated into the objective function via penalty terms, so the problem is presented in a form with no explicit constraints. The objective of a QUBO is a quadratic polynomial in these binary variables. The standard QUBO formulation is:

$$\min_{x \in \{0,1\}^n} x^T Q\, x, \tag{4.1}$$

where Q is an $n \times n$ real-valued upper-triangular (or symmetric) matrix containing weights for variable interactions. Expanding this quadratic form yields:

$$\min_{x \in \{0,1\}^n} \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}\, x_i x_j \tag{4.2}$$

Since $x_i^2 = x_i$ for binary variables (i.e., x_i is either 0 or 1), purely linear terms can be embedded within the quadratic framework by setting $Q_{ii} = q_i$:

$$\min f(x) = \sum_i q_i x_i \; + \; \sum_{i<j} Q_{ij}\, x_i x_j, \tag{4.3}$$

where:

- $q_i = Q_{ii}$ are linear coefficients representing individual variable contributions,
- Q_{ij} (for $i \neq j$) are pairwise interaction coefficients, defining how two binary variables influence the objective.

The task is to find the binary vector x^* that minimizes $f(x)$.

4.1.1 Why QUBO Matters

The QUBO framework is widely used because:

1. **Generalized Combinatorial Modeling:** Many NP-hard problems (e.g., MAX-CUT, Graph Partitioning, Boolean Satisfiability) can be expressed as QUBOs [38].
2. **Direct Quantum Compatibility:** Quantum annealers (such as D-Wave) solve problems in an Ising model form, which can be directly mapped to QUBO matrices [37].
3. **Specialized Classical Solvers:** Even outside quantum computing, several neuromorphic architectures and classical solvers are optimized for QUBO problems [39].

Despite its flexibility, solving a QUBO is NP-hard, as it generalizes well-known hard combinatorial problems. However, heuristic and metaheuristic methods (simulated annealing, genetic algorithms, etc.) enable practical solutions for large instances.

4.2 Higher-Order Unconstrained Binary Optimization (HUBO)

A HUBO model extends QUBO by allowing terms of degree higher than two in the objective function. Instead of just linear and quadratic terms, a HUBO formulation may include:

- Cubic terms: $x_i x_j x_k$
- Quartic terms: $x_i x_j x_k x_\ell$

- Higher-order interactions: $x_{i_1} x_{i_2} \dots x_{i_k}$

The general HUBO objective function can be expressed as:

$$H(x) = \sum_{i_1, i_2, \dots, i_k} a_{i_1 i_2 \dots i_k} \, x_{i_1} x_{i_2} \cdots x_{i_k}, \tag{4.4}$$

where each x_i is binary and the coefficients $a_{i_1 i_2 \dots i_k}$ represent interaction weights for multi-variable terms. The value k denotes the order of the highest-order term; QUBO is the special case where $k \leq 2$.

HUBO models are more expressive than QUBOs and arise naturally in problems with multi-variable dependencies, such as:

- Higher-order Markov Random Fields (MRFs) in machine learning
- Complex interaction models in computational physics and chemistry
- Operations research problems where rewards/costs depend on combinations of multiple binary decisions

While HUBO models provide a richer representation, they are even more computationally difficult than QUBO problems because:

1. Solving HUBO directly is intractable for large k (exponential scaling).
2. Most existing solvers and quantum hardware only support quadratic terms—necessitating a transformation from HUBO to QUBO.

4.3 From HUBO to QUBO – Problem Mapping

Because most combinatorial solvers (classical and quantum) are designed for QUBO problems, it is common to convert a HUBO into an equivalent QUBO. This transformation process is known as quadratization, and it introduces auxiliary binary variables to replace higher-order terms.

Consider the cubic term $x_i x_j x_k$ in a HUBO. We introduce an auxiliary binary variable z such that:

$$z = x_i x_j x_k$$

To ensure z correctly represents the product $x_i x_j x_k$, we introduce penalty constraints into the objective function, such as:

$$P(x, z) = M(z - x_i x_j x_k)^2 \tag{4.5}$$

where M is a large penalty coefficient. Expanding and simplifying this term introduces only quadratic interactions, converting the problem into a QUBO.

A fundamental result by Rosenberg (1975) [40] states that any pseudo-Boolean function (i.e., a binary polynomial objective) can be transformed into an equivalent quadratic form by introducing extra variables and constraints. While this transformation is always possible, it can significantly increase problem size, making it computationally expensive. Therefore:

- If a problem is naturally quadratic, formulating it directly as a QUBO is preferred to avoid introducing extra variables.
- If the problem contains high-order terms, one must carefully apply quadratization techniques, balancing accuracy, problem size, and solver compatibility.

For example, in space debris removal optimization, certain time-of-flight terms introduced high-order dependencies between decision variables. To solve this problem efficiently on available solvers, a HUBO-to-QUBO transformation was required, leveraging auxiliary variables and penalty terms.

Chapter 5

Quadratization Techniques

5.1 Classical Quadratization Methods

Quadratization is the process of transforming a Higher-Order Unconstrained Binary Optimization (HUBO) problem into a Quadratic Unconstrained Binary Optimization (QUBO) problem. This transformation is essential because most modern combinatorial solvers—including quantum annealers, Ising machines, and specialized classical solvers—are designed to handle at most quadratic terms. Quadratization ensures that a higher-order polynomial objective can be rewritten using only linear and quadratic terms, while maintaining the equivalence of optimal solutions.

The general quadratization process follows these key steps [41]:

1. **Identify higher-order terms:** List all polynomial terms of degree greater than two (e.g., cubic terms $x_i x_j x_k$, quartic terms $x_i x_j x_k x_\ell$, etc.). Each such term is a candidate for reduction.

2. **Introduce auxiliary variables:** Replace each higher-order term with a new binary variable that represents the product of certain original variables.

3. **Add penalty terms:** Modify the objective function by introducing penalty constraints to enforce consistency between the auxiliary variables and the original binary variables. These penalties are constructed so that they are minimized (zero or low cost) if and only if the auxiliary truly equals the product of the originals.

Once these steps are applied systematically, the problem is reduced to a purely quadratic form (QUBO) in an expanded variable space. The optimal solution of the transformed QUBO should ideally satisfy all the penalty constraints, ensuring it corresponds to a valid solution of the original HUBO.

5.1.1 Quadratization of a Cubic Term: A Concrete Example

Consider a cubic term $x_i x_j x_k$ in a HUBO problem. The goal is to replace it with a new auxiliary variable y while maintaining equivalence in the objective function.

1. **Introduce an Auxiliary Variable**

   Define a new binary variable $y = y_{ijk}$ that should ideally represent the product $x_i x_j x_k$.

2. **Replace the Higher-Order Term**

   Modify the objective function by replacing $x_i x_j x_k$ with y.

3. **Enforce Consistency with a Penalty Term**

   To ensure that y correctly represents $x_i x_j x_k$, add a penalty function that assigns a cost if $y \neq x_i x_j x_k$. A common penalty function is:

   $$P(y - x_i x_j x_k)^2$$

   where P is a large positive coefficient ensuring that the constraint is enforced in the optimal solution.

   Expanding this term:

   $$P(y^2 - 2y x_i x_j x_k + x_i^2 x_j^2 x_k^2)$$

   Since $y^2 = y$ and $x_i^2 = x_i$ for binary variables, this simplifies to:

   $$P(y - 2y x_i x_j x_k + x_i x_j x_k)$$

   This term is now quadratic, meaning the HUBO has been successfully converted to QUBO.

5.1.2 Rosenberg's Quadratization Method (1975)

A classical quadratization technique, proposed by Rosenberg (1975) [40], is widely used to reduce high-order binary interactions. Instead of using a direct squared-penalty function, Rosenberg introduced auxiliary variables that enforce consistency through alternative linearized constraints.

For example, to quadratize a cubic term $x_i x_j x_k$, we introduce an auxiliary y and impose the following transformation:

$$M(x_i x_j - 2x_i y - 2x_j y + 3y) \tag{5.1}$$

- If $y = x_i x_j$, the penalty evaluates to zero, ensuring a valid transformation.
- If $y \neq x_i x_j$, the penalty incurs a cost, discouraging infeasible assignments.

This technique efficiently reduces cubic or higher-order terms (by applying such transformations iteratively) without introducing excessive auxiliary variables, making it preferable in many applications.

5.1.3 Quadratization of Quartic and Higher-Order Terms

For a quartic term $x_i x_j x_k x_\ell$, a common approach is to reduce it in two steps:

1. Introduce an intermediate auxiliary variable z to represent $x_i x_j$, replacing the quartic term with $z x_k x_\ell$.
2. Introduce a second auxiliary y to represent $x_k x_\ell$, yielding a purely quadratic formulation zy.

This recursive reduction method generalizes to higher-degree terms, ensuring that any HUBO can be quadratized systematically.

5.2 Efficiency Considerations in Quadratization

While quadratization is mathematically guaranteed to work, it introduces new challenges:

1. **Increased Problem Size:** Introducing auxiliary variables expands the search space [41]. If a problem originally has n variables, quadratization may increase this to $n + m$, where m is the number of auxiliaries. This makes solving the QUBO more computationally expensive, both for classical algorithms (more variables to consider) and for quantum annealers (more qubits required).
2. **Penalty Strength Selection:** Choosing an appropriate penalty coefficient P is crucial. If P is too small, the solver may return infeasible solutions. If P is too large, it can distort the solution landscape and affect numerical stability.
3. **Avoiding Local Minima:** Some quadratization methods introduce spurious local minima, where the solver gets stuck in infeasible configurations. Careful tuning of penalty functions is necessary to mitigate this issue.

Thus, an active area of research is finding alternative quadratization approaches that reduce higher-order terms while minimizing side effects. Researchers explored alternative methods:

- **Minimal Auxiliary Variable Methods:** Some quadratization techniques seek to reduce the number of auxiliary variables, preserving a compact problem formulation [42].
- **Adaptive Penalty Scheduling:** Instead of using a fixed penalty coefficient P, some methods use an adaptive approach that gradually increases penalties during optimization [43].
- **Hypergraph-Based Quadratization:** Advanced techniques use hypergraph structures to systematically decompose higher-order interactions into chains of quadratic terms, reducing computational overhead [44].

In this thesis, a novel quadratization method was developed for the specific ADR optimization problem to handle a troublesome high-degree term efficiently – by exploiting problem-specific structure (the orbital mechanics context) to reduce polynomial degree with minimal overhead.

Chapter 6

Solvers and Optimization Algorithms

Once a mathematical model is defined—whether as an Integer Linear Program (ILP), Quadratic Unconstrained Binary Optimization (QUBO), or another formulation—the next step is solving it efficiently. For NP-hard combinatorial problems, exact methods such as brute-force search, branch-and-bound, and cutting-plane algorithms quickly become impractical as problem size grows due to their exponential worst-case complexity. Instead, heuristic and metaheuristic solvers are commonly employed, prioritizing scalability and speed over guaranteed optimality.

6.1 Heuristic and Metaheuristic Approaches

A heuristic is a problem-specific rule or algorithm that constructs a solution step by step, often using locally optimal choices [45]. For instance, a greedy heuristic for the Traveling Salesman Problem (TSP) might always select the nearest unvisited city. While these approaches are computationally efficient, they are prone to getting stuck in suboptimal solutions.

To address this, metaheuristics provide higher-level strategies that enhance search exploration and help escape local minima. Several widely used metaheuristics in combinatorial optimization include:

- **Simulated Annealing (SA):** Inspired by the annealing process in metallurgy, SA explores solutions by accepting both improving and worsening moves with a probability that decreases over time [19]. Early on, it allows large, disruptive moves to escape local optima; as the temperature parameter decreases, the search narrows around promising regions. SA is particularly well-suited for QUBO problems, where the energy landscape corresponds to the optimization objective. However, performance heavily depends on the cooling schedule, and poor parameter tuning can lead to premature convergence or inefficient exploration.

- **Genetic Algorithms (GA):** GA is based on natural selection principles, maintaining a population of candidate solutions that evolve over generations [20, 46]. Through mechanisms like crossover (recombination of solutions), mutation (random changes), and selection (favoring better solutions), the algorithm explores the search space broadly. GA is particularly effective for large, complex optimization problems like scheduling and routing. However, it requires careful tuning of mutation rates, population size, and selection pressure and can suffer from slow convergence if the population lacks diversity.

- **Tabu Search (TS):** TS enhances local search by using a memory-based mechanism to avoid revisiting recently explored solutions [47]. Each move is recorded in a tabu list, preventing immediate reversals and forcing the search toward unexplored regions. TS is particularly effective for problems where local search is strong but prone to cycles, such as graph-based optimization and QUBO problems, where flipping binary variables represents solution transitions. Despite its effectiveness, TS requires manual tuning of tabu tenure and can struggle with vast search spaces.

Other advanced metaheuristics include Ant Colony Optimization (ACO) [48], Particle Swarm Optimization (PSO) [49], Iterated Local Search (ILS), and hybrid methods that combine elements from multiple approaches. Beam Search (a truncated tree search) and Reinforcement Learning techniques have also been explored for combinatorial optimization, particularly in mission planning problems like Active Debris Removal (ADR).

6.2 Quantum Annealing (QA) and Hybrid Quantum Annealing (HQA)

Quantum optimization can be approached through two primary paradigms: the quantum circuit model [50] and adiabatic quantum computation (AQC) [51]. The quantum circuit model, widely used for algorithms like Shor's factoring algorithm and Grover's search, relies on sequences of unitary quantum gate operations applied to qubits. In contrast, AQC operates by continuously evolving a quantum system from the ground state of a simple Hamiltonian to the ground state of a final Hamiltonian that encodes the optimization problem.

This work focuses on Quantum Annealing (QA), a heuristic implementation of AQC that is tailored specifically for combinatorial optimization problems [52, 53]. While QA follows the adiabatic principle, it operates in a non-ideal regime where hardware limitations, thermal effects, and noise influence the computation.

Unlike an ideal AQC process, which requires infinitely slow evolution to guarantee optimality, practical quantum annealers are designed to find good solutions within finite annealing times.

6.2.1 How Quantum Annealers Work

Quantum Annealers (QAs) are specialized quantum devices optimized for solving Quadratic Unconstrained Binary Optimization (QUBO) problems [29]. They leverage two fundamental quantum phenomena:

1. Superposition, which allows the system to explore multiple solutions simultaneously.

2. Quantum tunneling, which enables the system to pass through energy barriers rather than climbing over them, reducing the likelihood of getting trapped in local minima.

A QA consists of a network of superconducting qubits, physically implemented using Josephson junctions [53, 54]. These qubits interact via programmable couplers, allowing the system to encode an optimization problem as a quantum Hamiltonian. Unlike classical bits, which are strictly 0 or 1, qubits can be in a superposition of both states, enhancing the exploration of the solution space.

6.2.2 Solving a QUBO Problem with a Quantum Annealer

The annealing process in QA follows these key steps:

1. **Initialization**

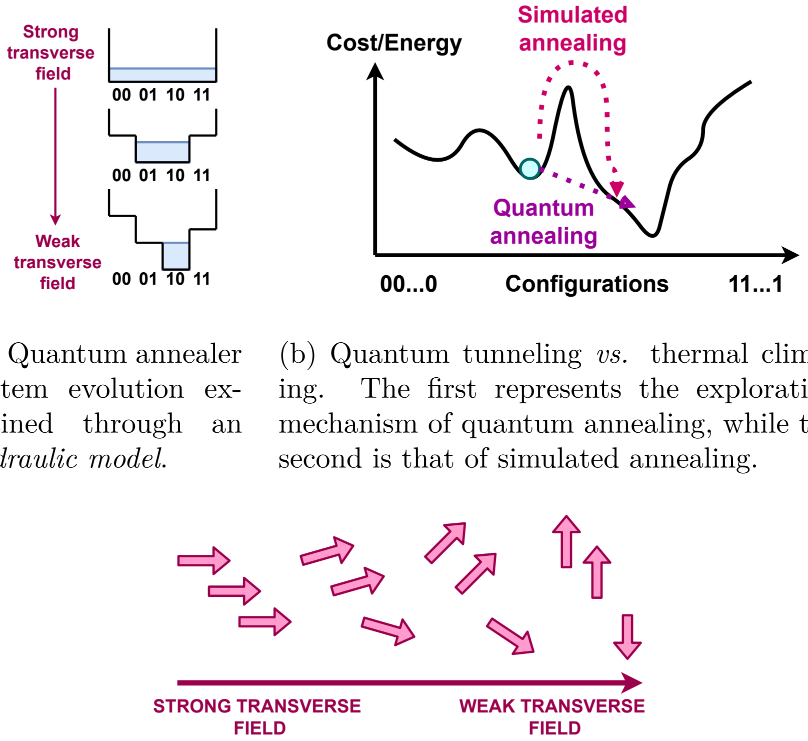
   The system begins in a uniform superposition state, where each qubit is equally likely to be in $|0\rangle$ or $|1\rangle$. This corresponds to the ground state of a simple transverse-field Hamiltonian, H_I.

2. **Adiabatic Evolution**

   The system's Hamiltonian is gradually transformed from the initial Hamiltonian H_I to the final Hamiltonian H_F, which encodes the optimization problem. This transition follows the time-dependent equation:

   $$H_{QA} = \varphi(t)H_I + (1 - \varphi(t))H_F, \tag{6.1}$$

   where $\varphi(t)$ smoothly decreases from 1 to 0 over the annealing time T_A. During this evolution, quantum tunneling helps the system avoid local optima by allowing transitions that would be difficult in classical methods.



(a) Quantum annealer system evolution explained through an *hydraulic model.*

(b) Quantum tunneling *vs.* thermal climbing. The first represents the exploration mechanism of quantum annealing, while the second is that of simulated annealing.

(c) Spins evolution during annealing. When a strong transverse field is applied, spins are in a superposition state. Reducing the transverse field, spins assume a spin-down or spin-up state to reach the lowest energy configuration.

Figure 6.1: Overview of quantum annealing.

3. **Measurement**

   At the end of the anneal, the system collapses into a classical binary configuration that represents a candidate solution to the optimization problem. This final state is read as σ^*, which corresponds to a binary vector solution ζ^*.

4. **Post-processing**

   Due to hardware noise and thermal fluctuations, the raw quantum output may require classical post-processing using methods like simulated annealing or tabu search to refine the solution further.

5. **Repetitions**

Since quantum annealing is probabilistic, the process is repeated multiple times to improve the likelihood of obtaining an optimal or near-optimal solution.

6.2.3 The Adiabatic Condition and Annealing Time Constraints

The effectiveness of QA relies on the Adiabatic Theorem, which states that if the Hamiltonian evolves slowly enough, the system remains in its lowest energy state. The minimum required annealing time T_A satisfies:

$$T_A \geq \frac{1}{\min_{t \in [0,T_A]} \Delta E(t)^2}, \tag{6.2}$$

where $\Delta E(t)$ is the energy gap between the ground state and the first excited state. If the evolution is too fast (i.e., T_A is too small), non-adiabatic transitions can occur, causing the system to settle in suboptimal excited states rather than the true global minimum.

In practice, a balance must be struck:

- Longer annealing times increase the probability of reaching an optimal solution but are constrained by hardware decoherence and practical runtime considerations.

- Shorter annealing times allow for faster computations but increase the risk of finding suboptimal solutions.

6.2.4 QUBO to Ising Model Mapping

Quantum annealers natively solve Ising model problems [55], which can be written as:

$$H_F = \sum_i h_i \sigma_i + \sum_{i,j} J_{ij} \sigma_i \sigma_j, \tag{6.3}$$

where:

- $\sigma_i \in \{-1, +1\}$ are spin variables (related to binary variables via $x_i = (1+\sigma_i)/2$.

- h_i are local qubit biases, influencing individual spins.

- J_{ij} represents coupling strengths between qubits, encoding pairwise interactions.

Since QUBO problems are naturally expressible in this Ising Hamiltonian form, they can be directly mapped onto quantum annealers like D-Wave systems.

6.2.5 Hybrid Quantum Annealing (HQA)

Due to the limited qubit count and connectivity in current quantum annealers, purely quantum optimization struggles with large, complex problems. Hybrid Quantum Annealing (HQA) combines classical pre- and post-processing with quantum optimization to improve scalability and solution quality [37].

One common HQA approach is problem decomposition:

1. The large problem is split into smaller subproblems that fit within the quantum hardware's constraints.
2. The quantum annealer solves each subproblem, and a classical solver stitches solutions together to form a global result.

This allows larger instances to be tackled while still benefiting from quantum speedups.

6.3 Classical vs. Quantum Optimization Methods: A Comparative Discussion

Optimization plays a central role in solving NP-hard problems, and over the decades, classical algorithms have evolved into powerful tools for tackling large-scale combinatorial optimization problems. However, with the emergence of quantum optimization methods, particularly quantum annealing (QA) and hybrid quantum-classical approaches (HQA), there is growing interest in whether quantum methods can offer an advantage over classical techniques. This section provides a comparative analysis of classical vs. quantum optimization, highlighting their respective strengths, limitations, and future potential.

6.3.1 Strengths of Classical Methods

Classical optimization techniques have been extensively refined and optimized, including exact algorithms, heuristics, and metaheuristics. Their maturity, flexibility, and wide availability make them the dominant approach for solving NP-hard problems today.

- **Maturity and Proven Performance:** Decades of research have led to highly optimized solvers for many NP-hard problems. For instance, integer linear programming (ILP) solvers, such as CPLEX, Gurobi, and SCIP, can handle thousands of variables using advanced techniques like branch-and-bound, branch-and-cut, and cutting planes. Specialized algorithms, such as Held-Karp dynamic programming for TSP or column generation for routing problems, further enhance efficiency.

- **Heuristics and Metaheuristics for Scalability:** Methods like Simulated Annealing (SA), Genetic Algorithms (GA), Tabu Search (TS), and Ant Colony Optimization (ACO) are problem-agnostic, highly customizable, and scalable to large instances. These approaches often find high-quality solutions in a reasonable time, even when exact methods become infeasible.
- **Direct Constraint Handling:** Classical solvers naturally handle hard constraints (e.g., ILP directly enforces feasibility through explicit constraints). In contrast, quantum annealing requires constraints to be embedded as penalty terms, which may lead to violations if not tuned properly.
- **Runs on Standard Hardware:** Classical solvers operate on widely available computing infrastructure, from laptops to high-performance clusters. They are not limited by qubit count, connectivity, or noise issues impacting quantum hardware.

6.3.2 Limitations of Classical Methods

Despite their strengths, classical methods face fundamental challenges, especially in scalability and search efficiency for large NP-hard problems.

- **Exponential Growth in Complexity:** Exact methods scale poorly—what works for 50 cities in TSP might fail for 100 cities due to factorial growth in solution space. Even heuristics struggle beyond certain problem sizes.
- **Local Minima Traps:** Classical algorithms can get stuck in local optima, especially for highly nonlinear or multimodal landscapes. While metaheuristics mitigate this, finding global optima remains difficult.
- **High Computational Cost for Large Problems:** Solving NP-hard problems often requires millions of iterations, making classical heuristics computationally expensive and computation time to grows exponentially. Additionally, problems with high-dimensional feasible regions can cause random or evolutionary searches to spend significant time in infeasible areas.

6.3.3 Strengths of Quantum Optimization

Quantum optimization, particularly quantum annealing (QA) and quantum approximate optimization algorithms (QAOA) [56], leverages quantum effects to explore the solution space fundamentally differently than classical methods.

- **Quantum Tunneling Overcomes Energy Barriers:** Unlike classical optimization, which requires “hill climbing” to escape local minima, quantum tunneling allows a system to pass through energy barriers, making it more likely to reach a better global solution.

- **Superposition Enables Parallel Exploration:** Quantum states allow the system to explore many possible solutions simultaneously, accelerating the search process.

- **Entanglement Encodes Correlations Efficiently:** Quantum correlations allow decision variables to be optimized jointly, capturing dependencies faster than iterative learning in classical heuristics.

- **Potential for Exponential Speedups:** While general speedups remain debated, certain spin glass problems show that quantum annealing avoids "critical slowing down" better than classical simulated annealing. This suggests that QA might outperform classical heuristics in specific problem instances.

- **Fast Solution Generation:** Quantum annealers can generate a diverse set of near-optimal solutions in microseconds to milliseconds. For example, in our ADR mission planning, the quantum annealer produced multiple low-energy solutions in under a second, giving decision-makers a variety of high-quality mission plans to choose from.

6.3.4 Limitations of Quantum Optimization

Despite its potential, quantum optimization faces significant practical challenges, particularly due to hardware limitations and the current state of quantum technology.

- **Hardware Constraints:** Current quantum processors have limited qubits, sparse connectivity, and noise issues, restricting their ability to solve large-scale problems natively.

- **Embedding Overhead:** Mapping a logical problem with N variables onto quantum hardware often requires many more physical qubits due to minor-embedding and chain qubits, significantly increasing problem size. If chains break, solutions become invalid.

- **No Guaranteed Optimality:** Like classical heuristics, quantum annealing does not guarantee the global optimum. It often requires thousands of runs, followed by post-processing to extract the best solution.

- **Constraint Handling is Indirect:** Since QA primarily solves unconstrained QUBO problems, hard constraints must be encoded as penalties. If penalties are not perfectly tuned, the QA may return infeasible solutions. In contrast, classical ILP solvers enforce strict feasibility.

6.3.5 Potential Impact and Outlook

If quantum optimization methods—whether improved QA hardware or fault-tolerant quantum gate models like QAOA—continue to scale, they could revolutionize NP-hard problem-solving.

A realistic near-term impact is that quantum optimization will augment classical solvers rather than replace them. We already see:

- **Hybrid Quantum-Classical Algorithms (HQA):** Classical solvers handle constraints, while quantum optimizers focus on combinatorial bottlenecks.
- **Quantum-Assisted Heuristics:** Quantum processors (QPUs) can rapidly generate diverse initial solutions, accelerating classical solvers.
- **Cloud-Based Hybrid Services:** Commercial providers offer hybrid solvers that automatically distribute workload across classical and quantum hardware.

If quantum optimization achieves scalability and reliability, it could extend the frontier of solvable problems, impacting fields like aerospace, finance, or logistics.

Part III

Problem Modeling and Quadratization Strategies

Chapter 7

Problem Description and Mathematical Formalization

7.1 The Space Debris Collection Problem

This article aims to address the problem of disposing and removing multiple space debris items using a single chaser spacecraft. As shown in Figure 7.1, the mission involves a sequence of maneuvers for the chaser:

- departing from a designated disposal orbit;
- rendezvousing and docking with each target debris;
- transporting and releasing each debris into the disposal orbit.

This process is repeated for all scheduled debris items. The disposal orbit is designed by the IADC guidelines, ensuring complete deorbit within 25 years. Compliance with international debris mitigation standards helps reduce long-term orbital congestion.

The debris population considered in this study is derived from real satellite debris data obtained from Two-Line Element (TLE) files, ensuring accurate orbital parameter representation. Since all debris items are in LEO with low eccentricity, their orbits can be approximated as circular. To optimize fuel efficiency, the selection of the target debris is restricted to objects with an orbital inclination similar to that of the chaser's disposal orbit, while variations in altitude and Right Ascension of the Ascending Node (RAAN) are possible.

The core issue is to obtain an optimal sequence of debris-clearing operations for minimizing propellant consumption and maximizing efficiency in a mission.

7.2 Definition of the Capture Pattern and Variables

For a mission designed to perform C capture operations on a set of D debris objects, the capture sequence is represented as:

$$d_0, \quad seq = \{d_1, d_2, d_3, ..., d_D\}, \tag{7.1}$$

where $d_i \in seq$ denotes the i^{th} debris object selected for capture, while d_0 represents the initial state of the chaser before any capture operation. The capture pattern consists of:

1. **Departure from the Disposal Orbit:**

   The chaser departs from its disposal orbit, characterized by an initial altitude a_{ril} and RAAN $\Omega_{c_{in}}$.

2. **RAAN Alignment via J_2 Perturbation:**

   To align its RAAN with that of the target debris ($\Omega_{d(i)}$, where $i = 1, \ldots, D$), the chaser exploits the Earth's oblateness effect, specifically the J_2 gravitational perturbation. This gradual drift in RAAN is used to minimize fuel consumption.

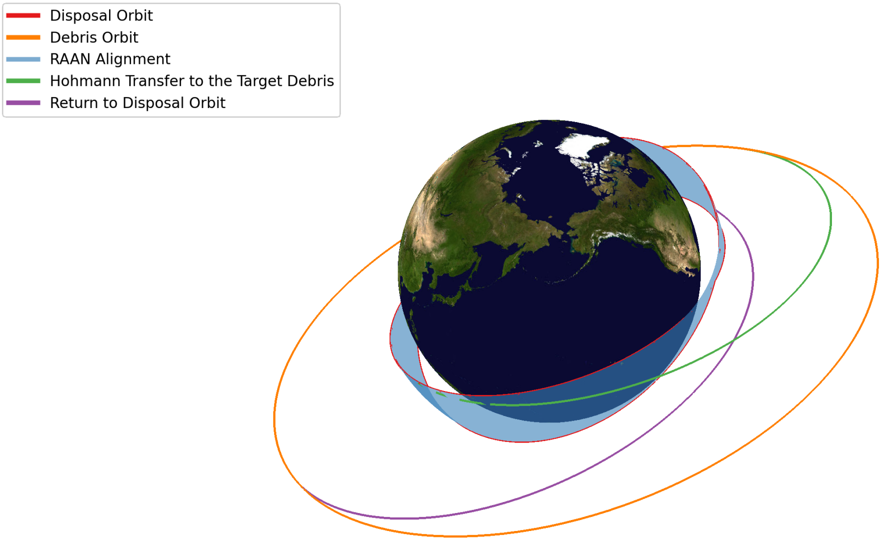


Figure 7.1: Mission maneuvers: depart from the disposal orbit, debris capture and return to the disposal orbit.

3. **Optimal Rendezvous Timing:**

   Once the chaser's RAAN aligns with that of the target debris, initiate a precisely timed waiting period. This ensures that, after executing a Hohmann transfer, the chaser arrives at the target orbit at the optimal position for rendezvous.

4. **Hohmann Transfer to the Target Debris:**

   The chaser executes a Hohmann transfer, an energy-efficient orbital maneuver, to move from its current orbit to the target debris orbit.

5. **Capture Operation:**

   Upon reaching the target orbit, the chaser executes a precise relative positioning maneuver, enabling the execution of the capture operation within a predefined period.

6. **Return to Disposal Orbit:**

   After capture, the chaser performs another Hohmann transfer to return to the disposal orbit.

7. **Debris Release:**

   Upon reaching the disposal orbit, the chaser remains in position for a specified period while releasing the captured debris.

8. **Cycle Repetition:**

   The process is repeated for the subsequent debris in the capture sequence until all target objects are processed.

In order to optimize the sequence of debris capture by minimizing operational costs (in terms of time and propellant consumption) while maximizing strategic benefits, the optimization problem includes three terms, described in the following.

7.3 TOF

The Time of Flight (TOF) represents the total time required to complete the capture of a debris object. For the capture of the i^{th} debris object in the selected sequence, TOF is expressed as:

$$\mathrm{TOF}_{(0,1,\ldots,i)} = t_{R(i-1,i)} + t_{W(0,1,\ldots,i)} + 2 \cdot t_{H(i)} + t_{cat} + t_{ril} \quad \forall i = 1,2,\ldots,C \tag{7.2}$$

7.3.1 RAAN Alignment Time t_R

t_R is the time required for the chaser's RAAN to align with that of the i^{th} debris object by exploiting the Earth's J_2 gravitational perturbation. The effect of the J_2 term in Earth's gravitational potential causes a deterministic, unidirectional variation of the RAAN, which is described by the following equation:

$$\dot{\Omega} = -\frac{3}{2} J_2 \left(\frac{R_E}{a(1-e^2)} \right)^2 n \cos(i) \,, \tag{7.3}$$

where:

- J_2 is the second zonal coefficient in the Legendre series approximating Earth's gravitational potential,
- R_E is Earth's equatorial radius,
- $n = \sqrt{\frac{\mu}{a^3}}$ is the mean motion,
- i is the orbital inclination.

Since the J_2-induced RAAN variation is unidirectional, its direction depends on the orbit's inclination:

- For prograde orbits ($i < 90°$), RAAN decreases over time:

$$t_{R(0,1)} = \frac{\text{mod}\ (\Omega_{d(1)} - \Omega_{c_{in}}, -2\pi)}{\dot{\Omega}},$$

$$t_{R(i-1,i)} = \frac{\text{mod}\ (\Omega_{d(i)} - \Omega_{d(i-1)}, -2\pi)}{\dot{\Omega}}$$

- For retrograde orbits ($i > 90°$), RAAN increases over time:

$$t_{R(0,1)} = \frac{\text{mod}\ (\Omega_{d(i)} - \Omega_{c_{in}}, 2\pi)}{\dot{\Omega}},$$

$$t_{R(i-1,i)} = \frac{\text{mod}\ (\Omega_{d(i)} - \Omega_{d(i-1)}, 2\pi)}{\dot{\Omega}}$$

- For polar orbits ($i = 90°$), the perturbation effect is null, meaning it cannot be used for RAAN alignment.

7.3.2 Hohmann Transfer Time t_H

t_H represents the time required to perform an orbital change using a Hohmann transfer maneuver. Whether moving to a higher or lower orbit, according to Kepler's third law, the transfer time is given by:

$$t_{H(i)} = \frac{1}{2}\sqrt{\frac{4\pi^2 \cdot a_{H(i)}^3}{\mu}} = \pi\sqrt{\frac{a_{H(i)}^3}{\mu}} \tag{7.4}$$

where $a_{H(i)} = \frac{a_{d(i)}+a_{ril}}{2}$ is the semi-major axis of the Hohmann transfer orbit.

7.3.3 Rendezvous Waiting Time: t_W

The waiting time t_W ensures that, after executing the Hohmann maneuver, the chaser arrives at the debris object's orbit at the correct position for rendezvous. The lead angle α_{lead} is the angular displacement between the initial position of the target and the rendezvous point, given by:

$$\alpha_{lead} = w_d \cdot t_H$$

Additionally, the final phase angle ϕ_f is the angular displacement between the chaser and the target at rendezvous:

$$\phi_f = \pi - \alpha_{lead}$$

To ensure correct timing, the chaser must wait such that:

$$\phi_f = \phi_i + (w_d - w_{ril}) \cdot t_W$$

Solving for t_W:

$$t_W = \frac{\phi_f - \phi_i}{w_d - w_{ril}}$$

Since the initial phase angle is $\phi_i = u_d - u_c$, we obtain:

$$t_W = \frac{\pi - w_d \cdot t_H - (u_d - u_c)}{w_d - w_{ril}}$$

If this expression yields a negative result, it is adjusted as follows, considering that $w_{ril} > w_d$ for every debris object (as the release orbit is always lower than the debris orbit):

$$t_W = \frac{\mathrm{mod}\ (\pi - w_d \cdot t_H - (u_d - u_c), -2\pi)}{w_d - w_{ril}} \tag{7.5}$$

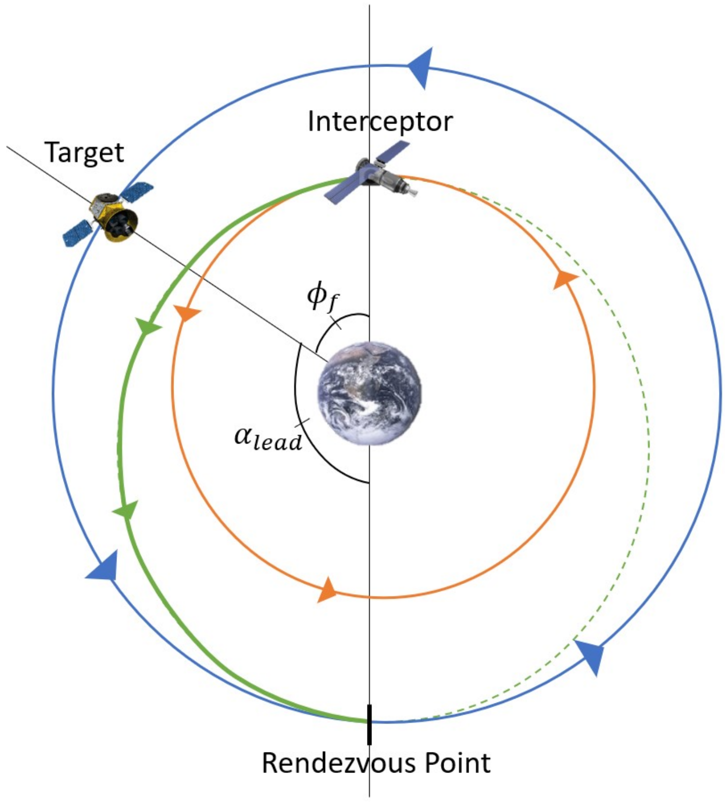


Figure 7.2: Depiction of a rendezvous maneuver between an interceptor and a target satellite. The diagram illustrates the phase angles ϕ_f and α_{lead} required for synchronization, showing both spacecraft trajectories and the rendezvous point. Image from Lynnane George's Open Educational Resource [57].

where u_d and u_c represent the arguments of latitude for the debris and the chaser, respectively, after RAAN alignment via J_2 perturbation. The J_2 perturbation also affects the argument of perigee (ω) and the mean anomaly (M), described by:

$$\dot{w} = \frac{3}{4} J_2 \left(\frac{R_E}{a(1-e^2)} \right)^2 n(5 \cdot cos^2(i) - 1) \tag{7.6}$$

$$\dot{M} = n \left(1 + \frac{3}{4} J_2 \left(\frac{R_E}{a(1-e^2)} \right)^2 \sqrt{1-e^2}(3\cos^2(i) - 1) \right) \tag{7.7}$$

Since we consider circular orbits ($e = 0$), the argument of perigee ω is undefined, and the true anomaly θ coincides with the mean anomaly M. Therefore, the variation in the argument of latitude u is influenced solely by M:

$$\dot{u} = \dot{M}$$

The chaser's argument of latitude is updated as follows:

$$u_{c(i-1,i)} = u_{c_{fin}(i-1)} + \dot{u} \cdot t_{R(i-1,i)}$$

$$u_{c_{fin}(i-2,i-1)} = u_{c_{fin}(i-2)} + \dot{u} \cdot t_{R(i-2,i-1)} + w_{ril} \cdot (t_{W(i-1)} + t_{ril}) + w_{d(i-1)} \cdot t_{cat} + 2\pi$$

$$\dots$$

$$u_{c_{fin}(0,1)} = u_{c_{in}} + \dot{u} \cdot t_{R(0,1)} + w_{ril} \cdot (t_{W(1)} + t_{ril}) + w_{d(1)} \cdot t_{cat} + 2\pi$$

where the final argument of latitude for previous captures recursively depends on earlier ones. For the debris argument of latitude:

$$u_{d(0,1,\dots,i)} = u_{d_{in}(i)} + w_{d(i)} \cdot (t_{R(i-1,i)} + \sum_{j=0}^{i-1} TOF_{(0,1,\dots,j)})$$

Thus, the waiting time for the i^{th} capture is:

$$t_{W(0,1,\dots,i)} = \frac{\mathrm{mod}\ (\pi - w_{d(i)} \cdot t_{H(i)} - (u_{d(0,1,\dots,i)} - u_{c(0,1,\dots,i)}), -2\pi)}{w_{d(i)} - w_{ril}} \tag{7.8}$$

This formulation accounts for all prior captures, emphasizing the sequential dependency in computing waiting times. This sequential dependency represents the bottleneck of this approach, making the optimization problem very complex. A solution to this issue is described in Section 9.2.

7.3.4 Capture Time t_{cat}

Once the chaser successfully reaches the debris, a specific period is allocated for the actual capture process. This phase includes: i) ensuring that the spacecraft is positioned and aligned with the debris, minimizing any relative motion, ii) deploying capture mechanisms (such as robotic arms or nets), iii) securing the debris, and iv) verifying that the capture is successful. The duration of this phase depends on the complexity of the capture system and the characteristics of the debris object.

7.3.5 Release Time t_{ril}

After securing the debris, it must be moved to the designated disposal orbit where it is either deorbited or stored safely. The release process includes positioning the debris in the correct disposal path and disengaging it from the chaser. Similar to the capture phase, this step requires careful execution to ensure that the debris follows the intended trajectory post-release.

7.4 Delta-v Δv

Delta-v (Δv) represents the instantaneous change in velocity that must be imparted to the chaser to perform the required orbital maneuvers. Since Δv is directly proportional to fuel consumption, it serves as a critical indicator in the optimization process. A Hohmann transfer is a commonly used maneuver that involves two propulsion impulses: the first accelerates the chaser into an elliptical transfer orbit with its periapsis at the initial orbit and its apoapsis at the target orbit; the second is applied at the apoapsis (for ascent) or periapsis (for descent) to circularize the orbit and match the debris's velocity. Using the vis viva equation, the velocity on the transfer orbit is:

$$v_{H,periapsis} = v_{H,p} = \sqrt{\mu(\frac{2}{r_{H,p}} - \frac{1}{a_H})}$$

$$v_{H,apoapsis} = v_{H,a} = \sqrt{\mu(\frac{2}{r_{H,a}} - \frac{1}{a_H})}$$

The total required for the Hohmann transfer is:

$$\Delta v = \begin{cases} |v_f - v_{H,a}| + |v_i - v_{H,p}|, & \text{if } r_i < r_f\ , \\ |v_f - v_{H,p}| + |v_i - v_{H,a}|, & \text{if } r_i > r_f\ , \end{cases} \tag{7.9}$$

Where $v_f = \sqrt{\frac{\mu}{r_f}}$ and $v_i = \sqrt{\frac{\mu}{r_i}}$. The total Δv for capture is given by the Hohmann maneuver required to reach the debris and return to the release orbit:

$$v_{Hcat,p(i)} = \sqrt{\mu(\frac{2}{a_{ril}} - \frac{1}{a_{H(i)}})}$$

$$v_{Hcat,a(i)} = \sqrt{\mu(\frac{2}{a_{d(i)}} - \frac{1}{a_{H(i)}})}$$

$$\Delta v_{cat(i)} = \left|\sqrt{\frac{\mu}{a_{d(i)}}} - v_{Hcat,a(i)}\right| + \left|\sqrt{\frac{\mu}{a_{ril}}} - v_{Hcat,p(i)}\right| \tag{7.10}$$

$$v_{Hril,p(i)} = \sqrt{\mu(\frac{2}{a_{d(i)}} - \frac{1}{a_{H(i)}})}$$

$$v_{Hril,a(i)} = \sqrt{\mu(\frac{2}{a_{ril}} - \frac{1}{a_{H(i)}})}$$

$$\Delta v_{ril(i)} = \left|\sqrt{\frac{\mu}{a_{ril}}} - v_{Hril,a(i)}\right| + \left|\sqrt{\frac{\mu}{a_{d(i)}}} - v_{Hril,p(i)}\right| \tag{7.11}$$

$$\Delta v_{tot(i)} = \Delta v_{cat(i)} + \Delta v_{ril(i)} \tag{7.12}$$

7.5 Desirability

Each debris object is associated with a desirability metric that quantifies its relative priority for capture. Although TLE files lack direct information regarding the mass or cross-sectional area of debris, they provide the drag term and other orbital parameters that—when processed via the SGP4 (Simplified General Perturbations Model 4) propagator [58]—allow for the estimation of decay times. In this study, the desirability of a debris object is defined as the time required for it to reach the disposal orbit naturally. Consequently, debris that would take a longer time to decay are assigned a higher desirability, thus prioritizing their capture over objects with shorter decay times.

7.6 Mathematical Formulation of Objectives and Constraints

The subsequent sections give a detailed description of the mathematical model, including the decision variables, important parameters, and the optimization function. Constraints are also formulated to guarantee the feasibility and effective use of the debris capture method.

7.6.1 Variables and Parameters

Decision Variable

We define the binary decision variable as follows: $x_{ij} \in \{0,1\}$:

- $x_{ij} = 1$ if debris j is captured in capture event i;
- $x_{ij} = 0$ otherwise.

Parameters

- $TOF_{(\cdot)}$: The total time associated with the complete sequence of captures. For a given sequence $(j_1, j_2, \ldots, j_C)$, the parameter $TOF_{(j_1,j_2,\ldots,j_C)}$ represents the total time required to execute all captures in that order. Note that, as previously discussed in Section X.X (t_W accounts for the cumulative effect of prior captures), the TOF parameter is calculated over the entire sequence and cannot be decomposed into components associated with individual x_{ij}.
- Δv_j: The total delta-v required to perform the two Hohmann maneuvers for capturing debris j. This parameter is directly linked to the decision to capture debris j during a specific event.

- des_j: The strategic desirability of debris j represents the benefit gained from its capture as described in Section X.X. This parameter is also directly incorporated into the corresponding term for x_{ij}.

Weighting Coefficients

C_T, C_V, and C_D are positive coefficients that balance the impact of TOF, Δv, and des within the objective function, respectively.

7.6.2 Objective Function

The model aims to optimize the capture sequence of space debris by minimizing operational costs (in terms of time and delta-v) while maximizing strategic benefits. Since the TOF parameter pertains to the entire sequence, its integration into the objective function is structured as follows.

Term Associated with TOF

For each capture sequence $(j_1, j_2, \ldots, j_C)$, where C represents the total number of capture events, we define:

$$C_T \cdot TOF_{(j_1,j_2,\ldots,j_C)} \cdot \prod_{i=1}^{C} x_{i,j_i}$$

This formulation ensures that $TOF_{(j_1,j_2,\ldots,j_C)}$ contributes to the objective function only when the selected debris items $j_1, j_2, \ldots, j_C$ are captured in sequence during events $1, 2, \ldots, C$ (i.e., when $x_{1,j_1} = x_{2,j_2} = \cdots = x_{C,j_C} = 1$).

Terms Associated with Δv and des

Since these parameters are associated with individual capture events, they contribute linearly:

$$C_V \sum_{i=1}^{C} \sum_{j=1}^{D} \Delta v_j x_{ij} \quad \text{and} \quad - C_D \sum_{i=1}^{C} \sum_{j=1}^{D} des_j x_{ij}$$

Complete Objective Function

By combining these terms, the final objective function is expressed as:

$$\min_x \quad C_T \sum_{(j_1,\ldots,j_C)\in S} TOF_{(j_1,\ldots,j_C)} \prod_{i=1}^{C} x_{i,j_i} \; + \; C_V \sum_{i=1}^{C} \sum_{j=1}^{D} \Delta v_j x_{ij} \; - \; C_D \sum_{i=1}^{C} \sum_{j=1}^{D} des_j x_{ij} \tag{7.13}$$

where S represents the set of all feasible debris capture sequences (i.e., all combinations that satisfy operational constraints).

7.6.3 Model Constraints

To ensure the operational feasibility of the solution, the model includes the following constraints:

1. **Unique Assignment Constraint:** Each debris item j can be captured at most once, ensuring the exclusivity of the action:

$$\sum_{i=1}^{C} x_{ij} \leq 1 \quad \forall j = 1, \ldots, D$$

2. **Operational Sequencing Constraint:** In each capture event i, at most one debris item can be selected, ensuring sequential execution:

$$\sum_{j=1}^{D} x_{ij} \leq 1 \quad \forall i = 1, \ldots, C$$

These constraints ensure that:

- No debris is captured more than once.
- No simultaneous captures occur within the same operational event.

7.6.4 Final Considerations

The model proposed succeeds in bringing together core features of the problem: selection at each event of the best debris to capture, balancing cost (time and delta-v) and benefits (desirability).

The objective function, in particular, through coefficients C_T, C_V, and C_D, allows modulation of each parameter contribution, and assignment and sequencing constraints allow maintenance of compliance with key operational requirements in any generated solution.

The model's salient features include:

- **The** TOF **term**, which depends on the overall sequence, is addressed through a high-order expression (product of sequence's x_{ij} values over its full sequence), emphasizing the sequential and cumulative nature of timing-related costs.
- **The** Δv **and** des **terms**, being tied to individual events, contribute linearly, making their integration into the objective function straightforward.

The inclusion of the TOF term adds complexity—requiring proper transformation into QUBO/HUBO-conformal form in follow-on rounds of optimization—but its form provides a good foundation for the expression of the debris capture problem in real and operational terms, in harmony with mission operational dynamics.

Chapter 8

Complexity Analysis and Theoretical Challenges

Building upon the formulation presented in Chapter 5, this chapter delves into the computational complexity of the objective function, analyzing how the solution space scales as the number of debris (D) and capture events (C) increases. The focus is on identifying the dominant computational bottlenecks and discussing potential strategies to mitigate them. The objective function is composed of two categories of terms:

- **Linear terms (Δv and des)**, which contribute linearly to the overall cost.

- **A high-degree polynomial term (TOF)**, which depends on the full capture sequence.

Here the explanation focuses on how the polynomial term induces exponential blowup in the solution space, with significant implications for problem solving.

8.1 Analysis of the Polynomial Objective Function and Its Complexity

From Subsection 7.6.2, we obtained the total objective function 7.13 as follows:

$$\min_x C_T \sum_{(j_1,\ldots,j_C)\in S} TOF(j_1,\ldots,j_C)\prod_{i=1}^{C} x_{i,j_i} + C_V \sum_{i=1}^{C}\sum_{j=1}^{D} \Delta v_j x_{ij} - C_D \sum_{i=1}^{C}\sum_{j=1}^{D} des_j x_{ij}$$

We therefore continue to perform a complexity analysis for each of its terms.

8.1.1 Analysis of the TOF Term

The polynomial term associated with TOF is:

$$f_{TOF}(x) = C_T\, TOF(j_1, \dots, j_C) \prod_{i=1}^{C} x_{i,j_i}$$

Complexity of Evaluating a Single Sequence

Considering a single sequence $(j_1, \dots, j_C)$, evaluating the polynomial term $\prod_{i=1}^{C} x_{i,j0_i}$ requires $O(C)$ operations , as the binary nature of x_{ij} makes each multiplication trivial. (since x_{ij} is binary, the computation itself has negligible cost). However, when extending this evaluation to all admissible sequences, the complexity escalates significantly due to the combinatorial nature of S.

Number of Admissible Sequences

Since each capture event must be uniquely assigned to debris, the total number of possible sequences is given by:

$$|S| = \frac{D!}{(D - C)!}$$

whose order of magnitude, for $D \gg C$, can be approximated as:

$$|S| = O(D^C)$$

Thus, an exhaustive evaluation of all possible sequences would result in an overall complexity of the TOF term of:

$$O(C \cdot D^C)$$

8.1.2 Analysis of the Linear Terms

The terms related to Δv and des sum linearly:

$$f_{\Delta v}(x) = C_V \sum_{i=1}^{C} \sum_{j=1}^{D} \Delta v_j x_{ij}, \quad f_{des}(x) = -C_D \sum_{i=1}^{C} \sum_{j=1}^{D} des_j x_{ij}$$

The computational cost of evaluating each of these terms is $O(C \cdot D)$, which is significantly lower than the polynomial component.

8.2 Impact of Increasing Debris and Capture Events on Computational Effort

This section analyzes how increasing the parameters D (number of debris) and C (number of capture events) affects the overall computational cost of the problem.

8.2.1 Effect of the Number of Debris (D)

Since the number of admissible sequences is:

$$|S| = \frac{D!}{(D-C)!}$$

For increasing values of D (with C fixed), the number of possible combinations increases rapidly. Asymptotically, if C is considered constant, we have:

$$|S| = O(D^C)$$

This means that even a moderate increase in D can lead to an explosion in the number of possible sequences, making solution space exploration computationally expensive.

8.2.2 Effect of the Number of Capture Events (C)

With D fixed, increasing the number of events C leads to exponential growth in the number of sequences, since:

$$|S| = \frac{D!}{(D-C)!} \approx O(D^C) \quad \text{for } C \ll D$$

In other words, increasing C by even one unit results in a multiplicative increase of D in the sequence count.

Practical Implications

This exponential dependency imposes practical limits on the problem dimensions that can be handled with an exact approach. For $D = 20$ and $C = 5$, for instance, the number of admissible sequences is on the order of $O(20^5) = O(3.2 \times 10^6)$; when $C = 6$, it reaches $O(20^6)$, exceeding 64 million possible configurations.

This increase highlights the need to resort to variable domain reduction strategies or heuristic/metaheuristic algorithms to handle realistically sized problems.

8.3 Overall Considerations

The analysis shows that the TOF term is the primary contributor to the combinatorial explosion of the solution space, leading to an asymptotic complexity of $O = (D^C)$. This makes the problem inherently NP-hard.

Effectively handling the TOF term requires a trade-off between computational feasibility and the accuracy of time cost modeling. Two primary approaches can be considered:

1. **Quadratization techniques**, which transform the high-degree polynomial terms into QUBO-compatible forms.

2. **Heuristic/metaheuristic methods**, such as simulated annealing or quantum annealing, to explore the solution space efficiently.

Chapter 9

Quadratization Strategies and Innovations

A key challenge in solving the debris capture optimization problem is the presence of high-degree polynomial terms in the objective function, particularly the TOF term. These terms hinder the use of quantum optimization methods, which require QUBO formulation. This chapter explores strategies for transforming the problem into a quadratic form, making it computationally feasible while preserving its essential structure. We will illustrate:

- Traditional quadratization techniques, with particular reference to substitution- and penalty-based methodologies;
- The proposed novel methodology, which simplifies the computation of TOF by approximating the waiting time term t_W using the synodic period, thereby reducing the problem's degree;
- A comparative analysis of both approaches in terms of computational performance and solution quality.

9.1 Traditional Quadratization of Polynomial Problems (Case >2 Captures)

Transforming a HUBO problem into a QUBO problem is a crucial step to leverage efficient optimization algorithms. In the context of this thesis, the '*to_qubo*' function from the Python library '*qubovert*' is used to automate the conversion by applying a quadratization procedure. Although the exact method implemented within '*to_qubo*' may vary, it typically relies on introducing auxiliary variables and adding penalty terms to ensure equivalence between the original product and the new formulation. The quadratization process (Section 5.1) follows a structured approach:

1. **Term Scanning:** The algorithm scans each term of the polynomial function. If the term is already quadratic (i.e., it contains at most two variables), it remains unchanged.

2. **Decomposition of Higher-Degree Terms:** For each term of degree $D > 2$ (e.g., a term of the form $\prod_{i=1}^{D} x_i$), the method proceeds to "decompose" the term through the following steps:

   - **Introduction of Auxiliary Variables:** A pair of variables, such as $x_i x_j$, is replaced with a new auxiliary variable z that must satisfy the condition $z = x_i \cdot x_j$.
   - **Addition of Penalty Terms:** To enforce the equivalence between z and the product $x_i x_j$, penalty terms are integrated into the objective function. An example of a penalty term is:

     $$\text{penalty} \times (x_i x_j - 2x_i z - 2x_j z + 3z)$$

     where the coefficient (called the "penalty") is appropriately chosen to "force" the new variable to assume the correct value.
   - **Iterative Process:** The algorithm proceeds iteratively or recursively until all terms are reduced to quadratic form.

This quadratization strategy, based on substitution and penalty addition, is well documented in the literature (e.g., Rosenberg, 1975 [40]; Boros & Hammer [41]) and is widely used to convert HUBO models into QUBO, for which numerous solution methods exist, including combinatorial optimization algorithms and quantum approaches.

9.2 Novel Generalized Quadratization Method

9.2.1 Context and TOF Problematic

In the original formulation 7.2, the total operation time TOF for a sequence of captures $(0, 1, \ldots, i)$ is expressed as:

$$\text{TOF}_{(0,1,\ldots,i)} = t_{R(i-1,i)} + t_{W(0,1,\ldots,i)} + 2 \cdot t_{H(i)} + t_{cat} + t_{ril} \quad \forall\, i = 1,2,\ldots,C$$

where:

- $t_{R(i-1,i)}$ represents the RAAN alignment time from the previous capture to the current one (thus a quadratic term),
- $t_{H(i)}$ is the time required to perform the Hohmann maneuver in capture i,
- t_{cat} and t_{ril} are the times associated with capture and release, respectively,
- $t_{W(0,1,\ldots,i)}$ is the waiting time necessary for the chaser to reach the debris at the right moment for rendezvous after the Hohmann maneuver.

The exact calculation of $t_{W(0,1,\ldots,i)}$ is particularly problematic as it requires knowledge of the entire sequence of previous captures, making the term's degree equal to the number of captures C. This high dependency results in significant computational complexity when formulating the problem (Section 8.1.1).

9.2.2 Approximation via the Synodic Period

The key challenge in reducing the complexity of TOF lies in the waiting time term t_W, which depends on the entire sequence of previous captures. To address this, we introduce a novel approximation: instead of computing t_W explicitly, we estimate it using the synodic period. The synodic period represents the time interval required for two orbiting bodies (in this case, the chaser and the debris) to realign in the same relative configuration. Operationally, it corresponds to the worst-case scenario: if the chaser has just missed an optimal departure opportunity, it must wait for a time equal to the synodic period to obtain the ideal alignment for rendezvous again.

This substitution implies:

$$t_{W(0,1,\ldots,j)} \approx T_{synodic,j}$$

where $T_{synodic,j}$ depends only on the debris j (i.e., on the variable x_{ij} associated with the current event) rather than on the complete sequence of captures. The primary advantage of this approximation is the reduction of the TOF term's degree:

- **The term** t_W is reduced from degree C to degree 1,
- **The overall** TOF **term** becomes quadratic (since $t_{R(i-1,i)}$ remains a quadratic term).

While this approximation results in a conservative estimate (i.e., an overestimation) of the waiting time, its impact on solution optimality is minimal, as the dominant contribution to TOF generally comes from the t_R term. Only in specific scenarios, such as multiple coplanar operations (when many debris share the same RAAN but have different altitudes), could the approximation's impact be more pronounced, though such cases are highly unlikely.

9.2.3 Advantages of the New Methodology

- **Reduction of Computational Complexity:** The simplification achieved by reducing the degree of t_W leads to a significant decrease in the number of required auxiliary variables and results in a more compact QUBO model.
- **Preservation of Solution Accuracy:** The synodic period approximation retains the primary set of optimal solutions, as the dominant term (t_R) remains unchanged, and the impact of t_W is relatively minor.
- **Applicability to Any Number of Captures:** The proposed methodology allows obtaining a quadratic formulation regardless of the number of captures, making the model scalable for realistic instances.

9.3 Comparative Analysis: Performance and Computational Efficiency

A fundamental difference between the two methodologies lies in the asymptotic complexity associated with handling high-degree terms:

- **Traditional Method:** Managing the TOF term exactly results in a degree equal to the number of captures C. Evaluating or converting such a term into a QUBO requires considering several combinations of $O(D^C)$, where D represents the number of available debris. This exponential complexity makes the problem inherently NP-hard and can significantly inflate the QUBO model.
- **New Methodology:** By replacing t_W with the synodic period, we achieve a quadratic formulation that dramatically reduces computational complexity from exponential ($O(D^C)$) to polynomial ($O(C \cdot D^2)$). This enables the use of QUBO-based optimization techniques on real-world problem instances, making large-scale debris capture scenarios computationally feasible.

Aspect	**Traditional Method**	**Proposed Method**
Degree of TOF term	$O(C)$	$O(2)$
Computational Complexity	$O(D^C)$	$O(C \cdot D^2)$
Accuracy	Exact, but computationally infeasible	Approximate, but retains key optimal solutions
Applicability	Limited to small-scale problems	Scalable to realistic mission sizes

Table 9.1: Comparative analysis of the traditional and proposed methods.

Chapter 10

Conversion to QUBO/HUBO Format

Bridging the gap between a mathematically quadratized model and a hardware-compatible QUBO/HUBO formulation is fundamental for leveraging quantum annealing. The challenge lies in encoding the problem in a way that maintains its computational tractability while respecting the constraints imposed by quantum hardware limitations. This chapter explores the methodology adopted to achieve this transformation, emphasizing the key decision to rely on a linear penalty term for quantum annealing, a choice driven by experimental insights into constraint violations in traditional "one-hot" encodings.

10.1 Mapping Techniques to QUBO/HUBO Format

The QUBO format is the standard for formulating combinatorial optimization problems intended for use on quantum hardware. As previously explained in Section 4.1, in this format, the objective function takes the form:

$$f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}$$

where $\mathbf{x}$ is the vector of binary variables and Q is a symmetric matrix containing the coefficients of the linear and quadratic terms. In cases where the original problem involves higher-degree monomials, a quadratization procedure is required (as described in Chapter 5), introducing auxiliary variables and penalty terms to ensure the correct reduction to QUBO/HUBO format. This mapping phase is crucial for leveraging quantum annealing techniques requiring a fully quadratic formulation.

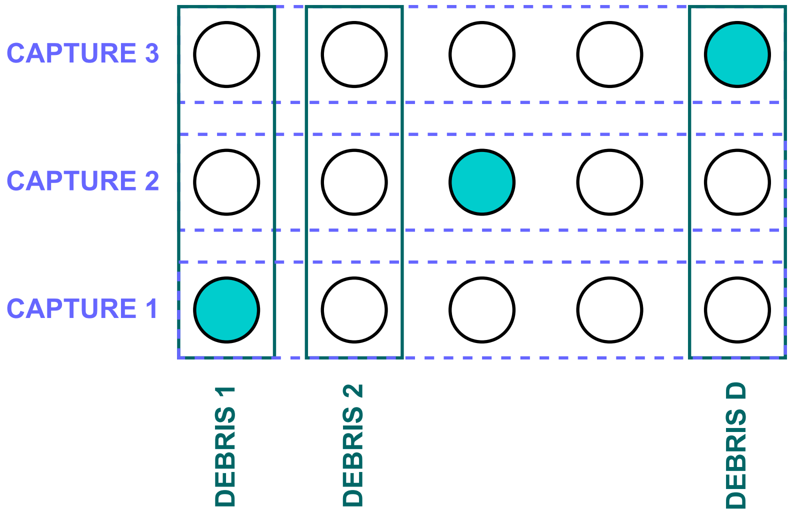


Figure 10.1: QUBO data structure, representing a valid solution. The highlighted circles are variables set to one, i.e. represent the debris-capture association.

10.2 Variable Structure and Costas Array Encoding

To encode decision variables, we employ a structure inspired by the Costas array, a well-established technique in permutation problems such as the Traveling Salesman Problem. In this rectangular encoding, the binary matrix $X = [x_{ij}]$ is organized as follows:

- **Rows** $(i = 1, \ldots, C)$: Represent capture operations, i.e., the events in which a selection is made.

- **Columns** $(j = 1, \ldots, D)$: Represent the available space debris.

This encoding imposes two essential constraints:

1. **Row Uniqueness:** each row must contain exactly one variable set to "1," formally expressed as: $\sum_{j=1}^{D} x_{ij} = 1 \quad \forall, i$.

2. **Column Uniqueness:** each column can contain at most one variable set to "1," i.e., $\sum_{i=1}^{C} x_{ij} \leq 1 \quad \forall, j$.

These constraints, analogous to the “one-hot” encoding used in permutation problems, have been translated into penalty terms in the QUBO model, as reported in the literature, for example, in the paper “Domain-Wall / Unary Encoding in QUBO for Permutation Problems” [59]. The traditional encoding, expressed in quadratic form, takes the following expression (where n corresponds to the size of the square Costas Array):

$$P_{2way1hot} = -\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} + \sum_{k=1}^{n}\sum_{i<j} (x_{ki}\, x_{kj} + x_{ik}\, x_{jk})$$

10.3 Definition of Penalty Terms and Variable Reduction Strategies

In the studied model, the final objective function to be minimized is formulated as:

$$f = C_T \cdot TOF + C_V \cdot \Delta v + C_D \cdot des + \lambda \cdot PenaltyTerm \tag{10.1}$$

Where the quadratic formulations of TOF, Δv, and des have already been presented in the previous chapters, while $PenaltyTerm$ consists of a combination of three components:

- **Term 1 – Binary Matrix Validity:** $P_{matrix} = \sum_{k=1}^{C}\sum_{j=1}^{D} x_{kj}$. This term penalizes the presence of an excessive number of “1”s in the matrix.
- **Term 2 – Row Uniqueness:** $P_{row} = \sum_{k=1}^{C}\sum_{j=1}^{D}\sum_{i=1}^{j-1} x_{ki}x_{kj}$. Ensures that no two “1”s appear in the same row.
- **Term 3 – Column Uniqueness:** $P_{column} = \sum_{k=1}^{D}\sum_{j=1}^{C}\sum_{i=1}^{j-1} x_{ik}x_{jk}$. Prevents two “1”s from appearing in the same column.

The overall $PenaltyTerm$ is defined as:

$$PenaltyTerm = \lambda_1 \cdot P_{matrix} + \lambda_2 \cdot P_{row} + \lambda_3 \cdot P_{column} \tag{10.2}$$

The three penalty components are inspired by the previously introduced “one-hot” approach, with a difference in the sign of the linear component.

Combining all terms, the complete objective function becomes:

$$\begin{aligned} f(x) &= f_{TOF} + f_{\Delta v} + f_D + \lambda_1 P_{\text{matrix}} + \lambda_2 P_{\text{row}} + \lambda_3 P_{\text{column}} = \\ &= C_T \sum_{(j_1,\dots,j_C)\in S} TOF_{(j_1,j_2,\dots,j_C)} \prod_{i=1}^{C} x_{i,j_i} + C_V \sum_{i=1}^{C}\sum_{j=1}^{D} \Delta v_j x_{ij} - C_D \sum_{i=1}^{C}\sum_{j=1}^{D} des_j x_{ij} + \\ &\quad + \lambda_1 \sum_{k=1}^{C}\sum_{j=1}^{D} x_{kj} + \lambda_2 \sum_{k=1}^{C}\sum_{j=1}^{D}\sum_{i=1}^{j-1} x_{ki}x_{kj} + \lambda_3 \sum_{k=1}^{D}\sum_{j=1}^{C}\sum_{i=1}^{j-1} x_{ik}x_{jk} \end{aligned} \tag{10.3}$$

10.3.1 Decision of Exclusive Use of P_{matrix} in QA

Experimental results revealed a critical issue when using traditional "one-hot" quadratic constraints in quantum annealing (QA): the annealer tended to converge towards high-energy configurations that violated the constraints, despite following a descending energy trajectory. The reason lies in the excessive coupler strength required to enforce row and column uniqueness, which distorted the energy landscape and led to invalid solutions.

To address this, we adopted a simplified penalty strategy, relying solely on the linear term P_{matrix} and eliminating P_{row} and P_{column} ($\lambda_2 = \lambda_3 = 0$). This approach offers two key advantages:

- **Reduction in the Number of Couplers:** Lowering quadratic terms significantly decreases the number of couplers required by the hardware, enabling better scalability for large debris capture problems.

- **Improved Exploration of Valid Solution Space:** Exclusive use of P_{matrix} has led to a dynamic annealing behavior where the optimal solution initially consists of all $x_{ij} = 0$. However, by properly adjusting the chain_strength parameter, it is possible to force the annealer to progressively explore configurations that satisfy the constraints: in fact, the qubit chain preserves the integrity of the logical variable and allows the system to gradually "scale" the number of "1"s present in the solution. In this way, the quantum annealer is able to identify valid solutions (i.e., those that respect the row and column uniqueness constraints) before ultimately converging to the optimal solution or an approximation of it.

10.3.2 Why a High Lambda Coefficient Worsens Annealing

A natural attempt to enforce constraints even in the QA case while continuing to use P_{row} and P_{column} could be to increase the λ coefficient associated with the entire $PenaltyTerm$. However, excessively increasing λ has been shown to have negative effects on the annealing process, as raising the coefficients λ_2 and λ_3 (when used) results in very high values in the couplers. Such high values can create a non-submodular energy landscape where local energy variations are too abrupt, making it difficult for the annealer to escape non-optimal local minima. In practice, excessively high weights of the penalty terms can "compress" the solution space, preventing proper exploration and favoring convergence to solutions with high energy. This issue led to the decision to eliminate the quadratic terms (P_{row} and P_{column}) in the QA case, using exclusively P_{matrix} and delegating constraint enforcement to the tuning of the chain_strength parameter.

10.3.3 The Role of Chain Strength and Embedding Management

In a quantum annealer, each logical variable is mapped onto a chain of physical qubits. If the coupling strength within the chain (chain strength) is too low, "chain-breaking" occurs, meaning that qubits representing the same variable take different values, leading to invalid solutions. In contrast, if the chain strength is too high, the system becomes rigid, reducing the annealer's ability to explore alternative configurations. Our approach, which relies solely on P_{matrix}, leverages precise tuning of the chain strength to guide the annealer through an effective exploration of the solution space. By gradually increasing the strength, we prevent the solution from prematurely converging to the configuration where all $x_{ij} = 0$ and promote a gradual transition in which the number of "1"s gradually decreases. This enables the identification, through a post-annealing analysis of the samples, of configurations that satisfy the constraints and have relatively low energy, facilitating the selection of the optimal solution.

10.4 Comparison Between QA and HQA/SA Strategies

The penalty strategies adopted differ depending on the platform used:

- **Quantum Annealer (QA):** In the QA case, the choice to use only the linear term P_{matrix} (with $\lambda_2 = \lambda_3 = 0$) led to a significant reduction in the number of required couplers and a more compact embedding. Although this results in an initially trivial solution (all $x_{ij} = 0$), tuning the chain strength allows the annealer to progressively explore the valid solution space, from which the lowest-energy solution will be selected. The simplified approach used for QA facilitates the resolution of problems with a larger number of debris.

- **Hybrid Quantum Annealing / Simulated Annealing (HQA/SA):** In these cases, the full use of constraints (including P_{row} and P_{column}) is more manageable, as hybrid or simulated annealing algorithms can tolerate greater complexity in the QUBO model. However, even in these cases, adding too many penalty terms increases the number of couplers, which can limit the scalability of the problem.

Part IV

Implementation, Experimentation, and Comparative Analysis

Chapter 11

Quantum Annealing Implementation

11.1 Hardware Description

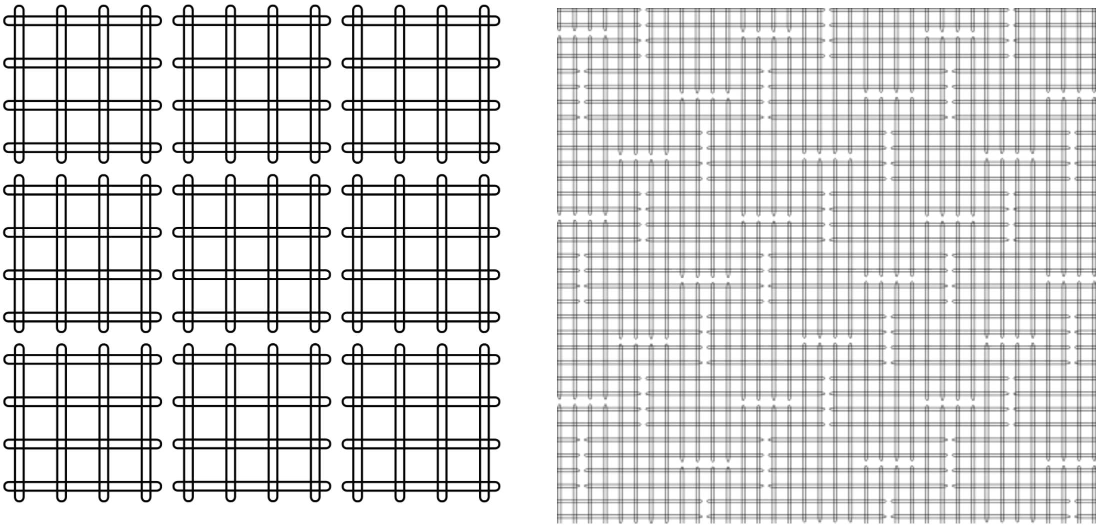
Optimizing multi-target Active Debris Removal (ADR) missions poses complex combinatorial challenges, often characterized as NP-hard. In recent years, quantum computing—and specifically quantum annealing—has emerged as a promising computational paradigm capable of tackling such challenges more efficiently than classical methods. In this section, the D-Wave quantum annealer is explored as a potential solution for large-scale trajectory optimization in ADR mission planning.

11.1.1 Quantum Annealer Architecture

The D-Wave quantum annealer [37] is engineered to exploit the principles of quantum mechanics to solve combinatorial optimization problems. Its architecture is based on superconducting flux qubits that are maintained at cryogenic temperatures (approximately 16 mK) inside a dilution refrigerator. The low-temperature environment is crucial for ensuring quantum coherence and minimizing thermal noise during computation.

Two main generations of D-Wave hardware are notable in literature: the earlier systems based on the Chimera topology and the more recent systems featuring the Pegasus topology. In the current generation (e.g., D-Wave Advantage), the hardware is characterized by:

- **Qubit Count and Layout:** The device comprises thousands of superconducting qubits arranged in a Pegasus graph (Figure 11.1b). This topology significantly enhances qubit connectivity compared to the earlier Chimera design (Figure 11.1a).

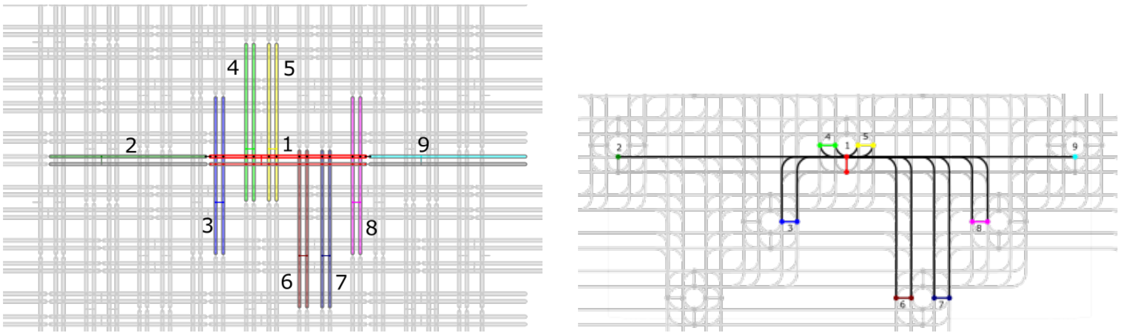


(a) Chimera topology used in earlier generations of D-Wave quantum annealers. Each unit cell contains eight qubits arranged in a bipartite structure, enabling local and long-range connections. Image sourced from D-Wave's official documentation [37].

(b) Pegasus graph representation illustrating the qubits and their couplings. Pegasus topology offers significantly increased connectivity compared to Chimera, enabling more complex QUBO embeddings. Image sourced from D-Wave's official documentation [37].

Figure 11.1: D-Wave QPU Topologies

- **Superconducting Circuitry:** The qubits are implemented as superconducting loops interrupted by Josephson junctions. These junctions provide the necessary non-linearity, allowing each qubit to represent a two-level quantum system ($|0\rangle$ and $|1\rangle$ states) with the possibility of superposition.
- **Connectivity and Couplers:** Each qubit is connected to multiple others through programmable couplers (Figure 11.2) that enable the encoding of interaction strengths (i.e., the Ising spin-spin coupling terms). These couplers encode the quadratic coefficients and enforce correlations between binary variables. The Pegasus topology typically provides each qubit with up to 15 or more couplings, which facilitates embedding larger and more complex QUBO problems.
- **Flux Biasing and Control:** Precise flux bias control ensures that qubits remain in their intended states during the annealing process. This control is crucial for maintaining the adiabatic evolution from the initial superposition



(a) Coupling graph representation.

(b) Roadway layout view.

Figure 11.2: Visualization of the Pegasus topology couplers in D-Wave quantum annealers. (a) shows the coupling graph with all possible connections between qubits, highlighting the high connectivity of Pegasus compared to earlier architectures. (b) presents the roadway layout, clarifying the structured routing of couplers and the physical arrangement of qubit connections. Images sourced from D-Wave's official documentation [37].

state to the final ground state that encodes the optimal solution.

- **Annealing Process:** Quantum annealing is achieved by slowly varying the Hamiltonian from an initial driver Hamiltonian H_I to the final problem Hamiltonian H_F. The process leverages quantum tunneling to escape local minima, with the minimum annealing time determined by the adiabatic theorem (Section 6.2). This controlled evolution is central to obtaining low-energy, near-optimal solutions.

11.1.2 Hardware Specifications and Performance

The operational specifications of the D-Wave quantum annealer [37] are optimized to support large-scale optimization problems:

- **Operating Temperature:** ~16 mK, maintained via a dilution refrigerator.
- **Qubit Count:** Modern systems such as the D-Wave Advantage offer over 5000 qubits.
- **Coupler Density:** The Pegasus topology allows for an increased number of couplers per qubit, which is critical for mapping complex QUBO problems with many variables and higher connectivity requirements.

- **Chain Embedding:** Due to limited qubit connectivity, logical variables may be represented by chains of physical qubits. The proper tuning of chain strength is crucial to preserving the integrity of the logical qubit and ensuring that the final measured state corresponds to a valid solution. If the chain strength is too weak, "chain breaking" may occur, leading to inconsistent representations; if too high, the system's ability to explore alternative configurations is impaired.

- **Precision and Noise:** Despite inherent noise and decoherence challenges in quantum hardware, careful calibration and improved qubit design have led to enhanced precision and reliability in solution sampling.

- **Scalability:** The increased connectivity and qubit count have made it feasible to tackle problems with hundreds of decision variables—an essential feature for solving the combinatorial explosion inherent in ADR mission planning.

11.2 Development Environment

11.2.1 Software Tools and Frameworks

The successful implementation of quantum optimization solutions on D-Wave hardware relies on a sophisticated software ecosystem. Central to this environment is the D-Wave Ocean Software Development Kit (SDK), a Python-based framework that simplifies the process of formulating and submitting QUBO problems to the quantum annealer.

Key components of the development environment include:

- **Ocean SDK:** This toolkit provides libraries such as *dimod* for defining binary quadratic models and QUBO formulations, as well as *dwave-system* for interfacing with the quantum hardware. Ocean SDK abstracts the complex embedding process, allowing researchers to focus on problem formulation rather than low-level hardware details.

- **Problem Formulation Libraries:** The SDK includes tools to convert optimization problems into a QUBO format. Functions for problem decomposition, variable embedding (using tools like minorminer), and constraint handling are provided. These features are critical when addressing high-degree polynomial terms present in the ADR optimization model.

- **Hybrid Solvers:** In addition to direct access to the quantum processing unit (QPU), the development environment supports hybrid solvers that combine classical and quantum optimization. These hybrid approaches (e.g., D-Wave's Leap Hybrid Solver) allow large or complex problems to be partitioned into subproblems, with the QPU solving the most computationally challenging parts while classical algorithms refine the overall solution.

- **Classical Optimization Libraries:** For benchmarking purposes, classical solvers—including Simulated Annealing (SA), Tabu Search (TS), and Genetic Algorithms (GA)—are implemented using *dwave* and *pymoo* [60] libraries in Python. These solvers are configured to solve the same QUBO formulation to enable a direct comparison of performance metrics.

11.2.2 Programming and Workflow Integration

The typical workflow for a quantum optimization task using the D-Wave system is as follows:

1. **Problem Modeling:** The ADR optimization problem is first modeled in mathematical terms. This includes defining the QUBO formulation that encapsulates both the objective function (minimizing fuel consumption and mission duration) and the necessary constraints (e.g., one debris per capture maneuver).

2. **Embedding:** Using the Ocean SDK, the problem is embedded onto the physical qubit connectivity graph. The minorminer tool automatically finds a mapping of logical variables onto physical qubits, considering the specific constraints of the Pegasus topology.

3. **Submission and Annealing:** The QUBO problem is submitted to the D-Wave QPU. The annealing schedule is configured (with parameters such as annealing time, chain strength, and repetition count) to maximize the probability of reaching a low-energy state.

4. **Post-Processing:** After annealing, the results are retrieved and subjected to classical post-processing. This step includes filtering out invalid solutions (those that violate constraints) and refining the best solutions using classical metaheuristic algorithms if necessary.

5. **Hybrid Optimization:** In cases where the problem size exceeds the QPU's direct capacity, a hybrid approach partitions the problem. The QPU tackles the core combinatorial challenge, while classical solvers handle the less computationally intensive parts of the model.

The integration of these tools enables a robust, iterative optimization process that leverages both quantum and classical computational strengths.

11.3 Dataset Handling

11.3.1 Source of Satellite Debris Data

Satellite debris data is primarily sourced from Two-Line Element (TLE) files, which provide the orbital parameters of space objects. TLE files are maintained by organizations such as NORAD and are publicly accessible via repositories like CelesTrak [61]. In this study, TLE data corresponding to known debris events (e.g., the collision between Iridium 33 and Cosmos 2251) serve as the input dataset for mission planning. Key parameters extracted from TLE files include:

- **Orbital Inclination (i):** Crucial for filtering debris that reside in similar orbital planes.
- **Semi-major Axis (a) and Eccentricity (e):** Although real orbits have eccentricity, debris in LEO are approximated as having circular orbits for the purpose of this model.
- **Right Ascension of the Ascending Node (RAAN, Ω):** Essential for computing the RAAN alignment time t_R.
- **Mean Motion (n) and Epoch Data:** Used in propagators such as SGP4 to estimate orbital evolution.

11.3.2 Data Filtering and Preprocessing

The raw TLE dataset undergoes several preprocessing steps:

- **Altitude Filtering:** Debris objects are filtered to ensure that only those above a designated minimum altitude (e.g., above 550 km) are considered. This is necessary because the disposal orbit is defined as being around 550 km to ensure compliance with IADC guidelines.
- **Inclination Filtering:** Debris with orbital inclinations within a narrow band (e.g., ±5° of the mean inclination) are selected to minimize the need for large inclination changes during the mission. Since the debris in our study largely originated from the same satellite breakup event, this filtering step minimally affected the overall dataset.
- **RAAN Consistency:** Objects with RAAN values that are significantly misaligned with the chaser's operational parameters are excluded.
- **Desirability Ranking:** Each debris object is assigned a desirability score based on its estimated natural decay time. Objects with longer decay times receive higher scores, reflecting their priority in removal operations.

The refined dataset consists of D debris objects, which serve as the columns in the binary decision matrix X.

Furthermore, in each scenario considered, the initial position and orbit of the chaser—defined by its argument of latitude and RAAN—were generated randomly. This approach allowed us to account for a broader range of possible mission configurations and prevented the optimization process from converging to the same solution across all cases. By varying these orbital parameters, we ensured a more diverse and robust assessment of feasible mission trajectories.

To standardize mission parameters and maintain consistency across scenarios, we imposed constant arbitrary durations for key mission phases. Specifically, the capture phase was set to a duration of 2 hours, while the release phase was set to 1 hour. These standard timeframes allow for uniform analysis of mission performance and operational feasibility across different scenarios.

11.3.3 QUBO Formulation from Debris Data

The processed satellite debris data is transformed into a QUBO problem that balances multiple mission objectives:

- **Objective Function Components:** The QUBO formulation incorporates terms representing the total delta-v (Δv) required for maneuvering, the overall time of flight (TOF) for each capture operation, and a desirability metric based on the debris' natural decay times (Subection 7.6.2).
- **Constraint Incorporation:** Assignment constraints ensure that each debris object is captured only once and that the capture maneuvers do not overlap. These constraints are embedded in the QUBO using penalty terms, carefully weighted to maintain the balance between feasibility and solution quality (Section 10.3).
- **Quadratization Techniques:** Since some terms (those related to TOF) are non-quadratic, a quadratization method is employed (Section 9.1). The novel method that approximates higher-order terms through the use of the synodic period is also used, thereby reducing the problem to a native quadratic form without introducing excessive auxiliary variables (Section 9.2).

This transformation from TLE data to a binary optimization problem is critical, as it bridges the gap between the raw orbital data and the quantum annealer's requirement for a QUBO input.

11.4 Overall Workflow

The methodology adopted in this work harnesses the complementary strengths of quantum and classical computing to address the multi-target Active Debris Removal

(ADR) optimization problem. Quantum annealing excels in rapidly exploring the vast combinatorial solution space of the QUBO model, while classical computing ensures precise data handling, constraint verification, and orbital dynamics simulation. The integrated workflow is structured as follows:

1. **Preprocessing:** Satellite debris data from TLE files is cleaned, normalized, and encoded into a binary format.

2. **QUBO Construction:** The multi-objective ADR mission planning problem is formulated as a QUBO, integrating objectives (minimizing TOF and Δv, maximizing debris desirability) and constraints.

3. **Quantum Optimization:** The QUBO problem is embedded and submitted to the D-Wave quantum annealer via the Ocean SDK. The annealer performs the quantum annealing process, returning candidate solutions.

4. **Post-Processing:** Classical algorithms filter and refine the quantum-generated solutions, ensuring all operational constraints are satisfied.

5. **Simulation Feedback:** The best candidate solution is fed into a classical orbital mechanics simulator to verify mission feasibility. Any discrepancies prompt adjustments to the QUBO formulation for subsequent iterations.

This seamless integration enables the system to harness the speed of quantum annealing while ensuring the practical applicability of the results through rigorous classical verification.

Chapter 12

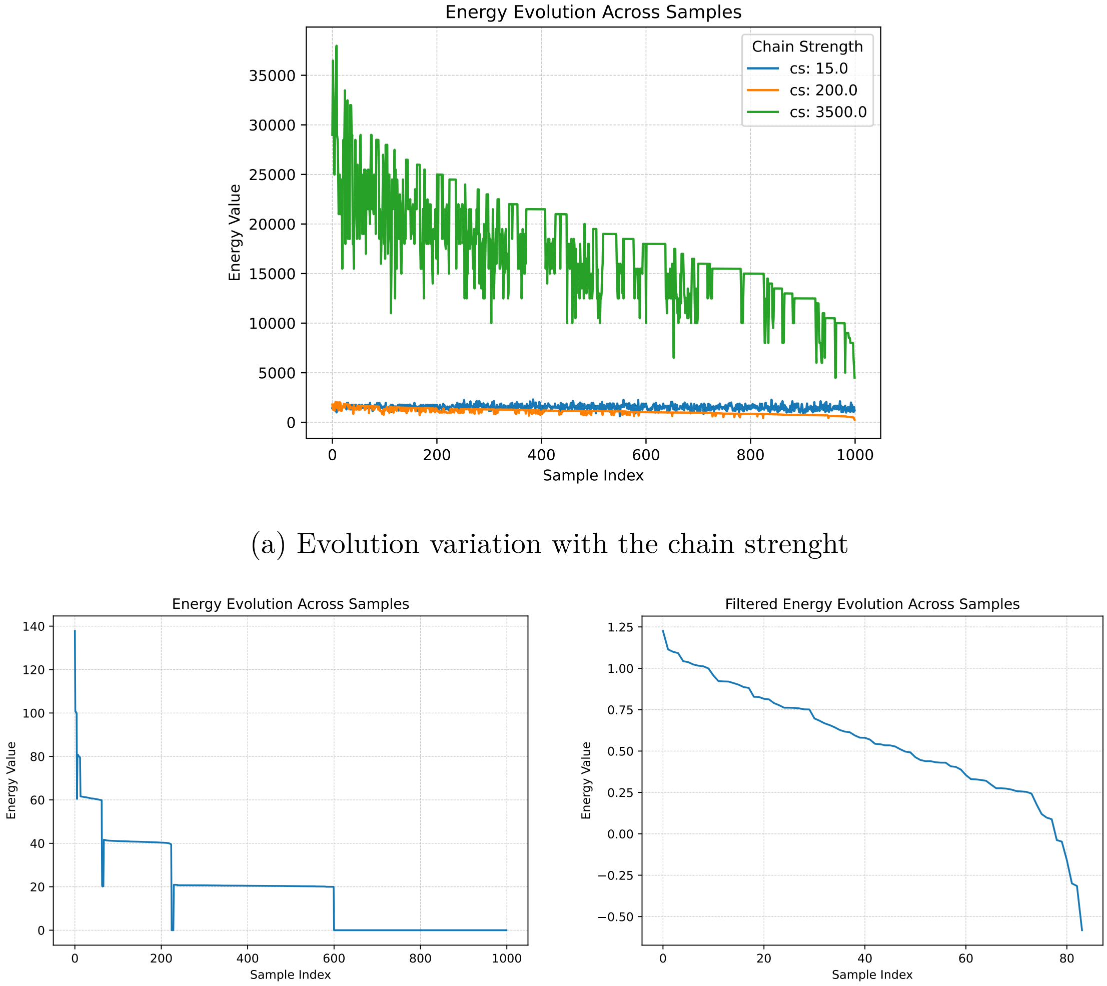
Comparative Analysis of Quantum and Classical Approaches

12.1 Quantum Annealing Constraint Management

The energy evolutions of QA reported in Figure 12.1 clearly demonstrate a significant improvement when using only the P_{matrix} penalty term. Initially, we evaluated the full set of constraints — specifically, P_{row} and P_{column} — across various chain strength settings in a scenario with 2 capture maneuvers and 50 debris objects from the Iridium 33 dataset (Figure 12.1a). Although the best energy value was achieved with a chain strength of 200, this configuration still yielded an invalid solution. Lowering the chain strength further resulted in highly oscillatory energy trends due to frequent chain breaks, while excessively high chain strength limits the solution space exploration; in this case, the energy decreased too slowly to reach any valid configurations. In essence, despite optimal tuning efforts, the conventional approach failed to enable effective exploration of the valid solution space, making it impossible to identify feasible — let alone optimal — solutions.

In contrast, using only the P_{matrix} penalty term yields a distinctive *staircase* search behavior, as shown in Figure 12.1b. In this approach, the annealer initially explores configurations with all the variables active (i.e., variables set to 1), then transitions sequentially to configurations with all active variables except one, and so on, until it reaches the trivial all-zeros solution. This stepwise exploration allows for a straightforward post-processing strategy: filtering the results to retain only those solutions with the correct number of active variables. Specifically, the solutions that satisfy the matrix constraints are maintained and that with the lowest energy is selected (Figure 12.1c).

This methodology allows for a more efficient exploration of the valid solution



(a) Evolution variation with the chain strenght

(b) Evolution considering only the P_{matrix} constraints

(c) Evolution considering only the P_{matrix} constraints and filtering solutions not satisfying the P_{row} and P_{column} constraints

Figure 12.1: QA energy evolutions, considering 2 capture maneuvers and 50 debris objects from the Iridium 33 dataset.

space compared to the traditional full-constraint approach. Even in relatively simple scenarios with few debris objects and capture maneuvers, the conventional method with full constraints consistently fails to lead any valid solutions when used with QA. On the other hand, the optimized P_{matrix}-based approach not only increases the likelihood of finding valid solutions but also enables more consistent identification of optimal solutions.

12.2 Solver Comparison and Scalability

C	D	Energy Value					Time (s)				
		QA	**HQA**	**SA**	**GA**	**TS**	**QA**	**HQA**	**SA**	**GA**	**TS**
2	15	**-0.56**	**-0.56**	**-0.56**	**-0.56**	**-0.56**	**0.12**	2.99	0.75	6.57	21.00
	20	**-0.65**	**-0.65**	**-0.65**	**-0.65**	**-0.65**	**0.21**	2.99	1.10	8.61	21.00
	25	**-0.58**	**-0.58**	**-0.58**	**-0.58**	**-0.58**	**0.14**	3.00	1.90	10.72	21.01
	50	**-0.58**	**-0.58**	**-0.58**	**-0.58**	**-0.58**	**0.19**	2.99	4.95	30.84	21.02
	75	-0.92	**-0.93**	**-0.93**	**-0.93**	**-0.93**	**0.22**	2.99	7.80	63.30	21.04
	100	**-1.37**	**-1.37**	**-1.37**	**-1.37**	**-1.37**	**0.25**	2.99	19.33	192.86	21.25
	114	-1.36	**-1.48**	**-1.48**	**-1.48**	**-1.48**	**0.27**	2.99	22.33	282.11	21.56
3	15	—	**-0.67**	**-0.67**	-0.29	**-0.67**	—	**2.99**	4.79	27.77	21.05
	20	—	**-0.64**	**-0.64**	0.07	-0.38	—	**2.99**	11.64	60.51	23.52
	25	—	**-0.94**	**-0.94**	0.60	-0.62	—	**2.99**	21.80	112.33	30.57
	50	—	-0.43	**-0.47**	0.75	—	—	**7.02**	91.57	2381.04	535.75
	75	—	-0.80	**-0.89**	—	—	—	**19.98**	400.57	5357.58	1253.38
	100	—	-0.66	**-0.88**	—	—	—	**43.95**	808.34	6632.49	3531.35
	114	—	**-1.21**	-1.06	—	—	—	**69.37**	1320.92	8696.00	27803.17
4	25	—	-0.07	**-0.21**	—	—	—	**8.43**	235.79	—	—
	50	—	0.69	**0.26**	—	—	—	**73.86**	3704.42	—	—

Table 12.1: Results obtained from the Iridium 33 datasets changing the number of capture maneuvers and debris objects.

The scalability of our approach is evidenced by the experimental results obtained for both the Iridium and Cosmos datasets (Table 12.1 and 12.2).

For the Iridium dataset, with 2 capture maneuvers, the QA solver consistently delivered competitive energy values while maintaining remarkably low computational times — even as the number of debris objects (D) increased from 15 to 114. For example, the QA solver's runtime increased only marginally from approximately 0.12 seconds to 0.27 seconds, whereas classical solvers such as SA and GA experienced substantial increases, with GA's time escalating from about 6.6 seconds to over 280 seconds.

In scenarios with 3 capture maneuvers, although QA results were not available, the HQA approach demonstrated stable energy performance with relatively low computational times compared to SA, GA, and TS. It is possible to notice that as the problem size grew, classical methods exhibited a dramatic rise in computation time: GA and TS, for instance, required several orders of magnitude more time (e.g., GA reaching over 2.300 seconds for $D = 50$ and TS exceeding 25.000 seconds for $D = 114$) compared to HQA.

The unavailability of QA results for 3 or more capture maneuvers stems from the nature of the problem formulation. Since the degree of the problem is equal to the number of capture maneuvers, cases with more than 2 captures required a quadratization process, which in turn introduced a substantial number of auxiliary variables.

Experimental results revealed that this excessive number of auxiliary variables hindered the QA solver's ability to effectively explore the valid solution space. Specifically, the newly introduced constraints from quadratization significantly restricted QA's search dynamics, leading to inaccurate and inefficient exploration of feasible solutions. As a result, the solver frequently produced constraint-violating solutions, making it impractical for cases involving more than 2 capture maneuvers.

However, this limitation is effectively addressed by the new quadratization method introduced in this work. By transforming the problem into a native quadratic form without excessive auxiliary variables, the new approach enables QA to handle problems with a higher number of capture maneuvers while preserving solution quality and computational efficiency. With this improved quadratization technique, QA can now be applied to scenarios involving larger numbers of captures without encountering the inefficiencies and constraints observed in the previous formulation.

C	D	Energy Value					Time (s)				
		QA	**HQA**	**SA**	**GA**	**TS**	**QA**	**HQA**	**SA**	**GA**	**TS**
2	25	**-1.00**	**-1.00**	**-1.00**	**-1.00**	**-1.00**	**0.20**	2.98	1.25	22	21
	50	**-0.81**	**-0.81**	**-0.81**	**-0.81**	**-0.81**	**0.16**	2.99	2.58	60	21
	75	-0.93	**-1.00**	**-1.00**	**-1.00**	**-1.00**	**0.22**	2.99	4.75	130	21
	150	-0.88	**-1.00**	**-1.00**	**-1.00**	**-1.00**	**0.27**	3.00	34	480	22

Table 12.2: Results obtained from the Cosmos 2251 datasets changing the number of capture maneuversand debris objects.

Similar trends are observed in the results from Cosmos dataset (Table 12.2). For 2 capture maneuvers, QA maintained a low runtime (ranging between 0.16 and 0.27 seconds) while delivering energy values equivalent to HQA, SA, GA, and TS. In contrast, classical solvers again showed significant increases in computation time as the number of debris objects increased — for example, GA required nearly 480 seconds for $D = 150$.

Table 12.3 reports the figures of merit for evaluating the mission performance obtained by the best QUBO solver in terms of final energy and solving time for the larger problem solved. It is possible to observe that the best solver is mainly a quantum-based solver.

Figures 12.2 and 12.3 show the execution time required to solve the debris collection problem using the Iridium 33 datasets, as a function of the number of debris objects. The results highlight the rapid growth in execution time for classical solvers as the problem size increases, while QA and HQA exhibit nearly constant performance, proving their scalability advantage.

Overall, these results demonstrate that the quantum annealing-based approaches, particularly QA and HQA, scale better with increasing problem complexity. The minimal increase in computational time with larger datasets, combined with competitive energy performance, suggests that quantum methods offer a promising

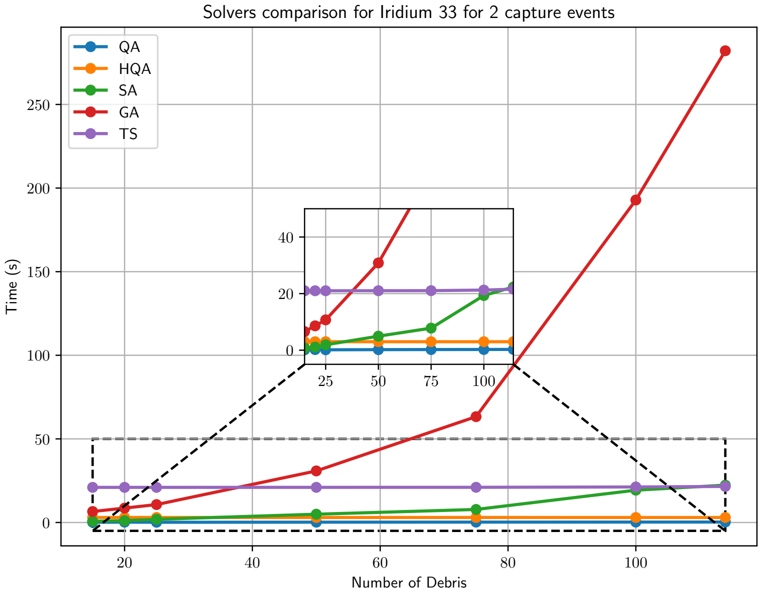


Figure 12.2: Time considering 2 capture maneuvers from Iridium 33 varying the number of debris.

pathway for tackling large-scale active debris removal mission planning.

Dataset	**CxD**	**Best Energy Value**				**Best Time**			
		Solver	**TOF (days)**	Δv	**des**	**Solver**	**TOF (days)**	Δv	**des**
Iridium	2x114	HQA	52.62	458.90	1.71	QA	52.62	442.86	1.55
	3x114	HQA	555.40	495.09	1.72	HQA	555.40	495.09	1.72
	4x50	SA	910.22	492.39	1.05	HQA	994.53	375.99	0.089
Cosmos	2x150	HQA	11.40	218.21	0.45	QA	105.48	175.69	0.25

Table 12.3: The best results obtained for the larger problems of the Iridium 33 and Cosmos 2251 dataset in terms of energy and time.

12.3 Advantage of the new Quadratization Method

Introducing our novel quadratization technique offers substantial advantages in both solution quality and computational efficiency. As reported in Tables 12.4, 12.5, 12.6, and 12.7 and in Figures 12.4, the new method significantly reduces the complexity of the objective function by transforming high-degree polynomial constraints into a native quadratic form without introducing auxiliary variables.

A key aspect of this improvement is illustrated in Table 12.4. For scenarios involving 3 and 4 capture maneuvers, the reported energy values are not directly

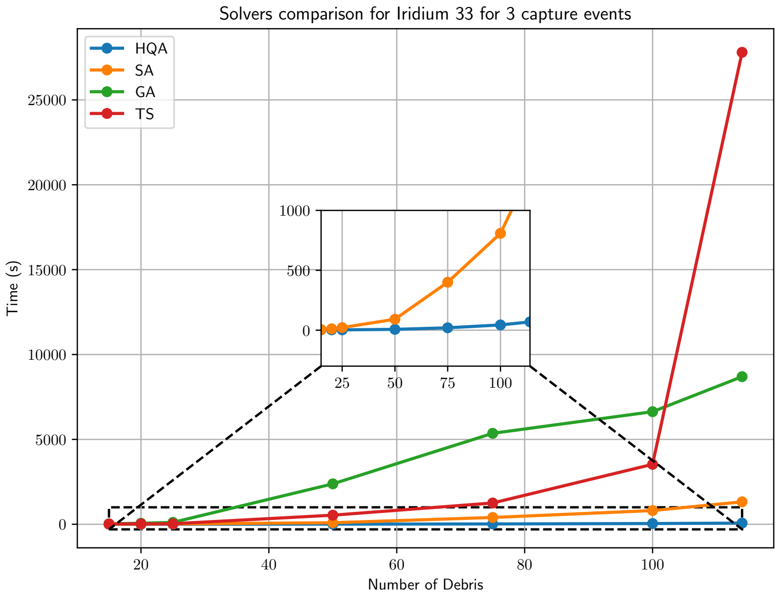


Figure 12.3: Time considering 3 capture maneuvers from Iridium 33 varying the number of debris.

C	D	Energy Value				Time (s)			
		HQA_quad	SA_quad	GA_quad	TS_quad	HQA_quad	SA_quad	GA_quad	TS_quad
3	15	**-0.67**	**-0.67**	**-0.67**	**-0.67**	3.00	**1.67**	9.25	21.00
	20	**-0.64**	**-0.64**	**-0.64**	**-0.64**	3.00	**2.63**	12.77	21.02
	25	**-0.94**	**-0.94**	**-0.94**	**-0.94**	**2.99**	3.53	17.82	21.01
	50	**-0.59**	**-0.59**	**-0.59**	**-0.59**	**3.00**	9.00	52.05	21.03
	75	**-1.25**	**-1.25**	**-1.25**	**-1.25**	**2.99**	23.88	108.51	21.07
	100	**-1.38**	**-1.38**	**-1.38**	**-1.38**	**3.00**	28.09	204.14	21.15
	114	**-1.55**	**-1.55**	**-1.55**	**-1.55**	**2.99**	34.15	269.05	21.53
4	25	**-0.76**	**-0.76**	**-0.76**	**-0.76**	**2.99**	3.29	21.44	21.01
	50	**-0.69**	**-0.69**	-0.66	**-0.69**	**3.00**	9.32	73.68	21.05
5	114	**-4.79**	-4.75	-4.24	-4.77	**2.99**	34.91	486.12	25.99
10	114	-8.69	-8.76	-6.92	**-9.41**	**3.25**	93.59	1107.29	49.09

Table 12.4: Results obtained from the Iridium 33 datasets changing the number of capture maneuvers and debris objects with the new quadratization method.

derived from the new function but instead obtained by evaluating the old function on the optimal solutions found using the new quadratization approach. This allows for a direct comparison between the two methodologies. However, for cases with 5 and 10 capture maneuvers, we report the energy values directly from the new method. This is because computing the energy using the old method becomes prohibitively time-consuming, and no corresponding results were available for direct comparison. The advantage of the new quadratization method is intuitively reported in terms of time, solution quality and composited efficiency in Figure 12.4a.

C	D	Energy Value				Time (s)			
		HQA_quad	**SA_quad**	**GA_quad**	**TS_quad**	**HQA_quad**	**SA_quad**	**GA_quad**	**TS_quad**
5	200	-4.37	-4.46	-4.47	**-4.52**	**3.00**	111.16	1489.61	42.40

Table 12.5: Results obtained from the Cosmos 2251 dataset with the new quadratization method.

The radar chart shows three normalized performance scores, each normalized in the range $[0, 1]$, where 1 represents the best performance. The scores are:

- **Solution Quality Score:** This metric measures the closeness between the energy of the obtained solution and the best-known energy — score one. For each run, a normalized quality gap (d) is calculated to measure how far the obtained solution is from the best observed quality. For each case, the minimum quality value (*best_val_min*) is identified, and a dummy value is used as a worst-case reference — score zero — for normalization. The dummy is defined as the energy of the trivial solution, i.e., collecting the debris in sequential order since we do not know the energy value of the valid solution with maximum energy. The gap is computed as:

$$d = \frac{best\_val - best\_val\_min}{dummy - best\_val\_min}$$

  Since a lower d indicates a better solution (with 0 being optimal), the final quality score is given by:

$$score\_d = 1 - mean\_d$$

  where $mean\_d$ is the average d for the optimization method.

- **Time Score:** This score evaluates execution time performance, where higher values indicate better efficiency. For each run, the time gap (t) is defined as:

$$t = \frac{tot\_time\_s}{min\_time} - 1$$

  where min_time is the fastest execution time observed for that file (so the fastest run has $t = 0$). Because lower t values are better, but we want scores where higher is better, we transform the values into scores between 0 and 1 using the formula:

$$score\_t(x) = \frac{max - x}{max - min}$$

  for each value x in the series of times (with max and min being the maximum and minimum values observed, respectively). This way, the best (lowest) time is mapped to 1 and the worst to 0.

- **Composite Efficiency Score:** This score provides an overall measure of performance by balancing both solution quality and execution time. It is derived from a combination of two normalized scores: one for solution quality and one for execution time, each contributing equally.

  $$score\_eff = 0.5 \times score\_d + 0.5 \times score\_t$$

  The highest efficiency score corresponds to the best trade-off between these two aspects, while lower scores indicate either poorer solution quality or longer execution times.

One of the most relevant advantages of the new quadratization technique is the drastic reduction in function assembly times, i.e. the time required for creating the QUBO. As evidenced in Table 12.7 and in Figure 12.4b, the new approach reduces these times by several orders of magnitude. For instance, in the Iridium dataset with 3 capture maneuvers, the function assembly time decreased from 0.5127 seconds (old method) to just 0.0241 seconds (new method) for 15 debris objects. Similarly, for 114 debris objects, the time was reduced from 278.65 seconds to just 1.1763 seconds. For 4 capture maneuvers, the improvements are even more pronounced: the assembly time dropped from 321.30 seconds to 0.0846 seconds for 25 debris objects and from 5949.25 seconds to just 0.3078 seconds for 50 debris objects.

Dataset	**CxD**	**Best Energy Value**				**Best Time**			
		Solver	**TOF (days)**	**Δv**	**des**	**Solver**	**TOF (days)**	**Δv**	**des**
Iridium	3x114	HQA_quad	554.02	687.38	2.60	HQA_quad	554.02	687.38	2.60
	4x50	HQA_quad	253.60	358.91	1.01	HQA_quad	253.60	358.91	1.01
	5x114	HQA_quad	880.43	1126.35	3.86	HQA_quad	880.43	1126.35	3.86
	10x114	TS_quad	375.38	1737.30	5.10	HQA_quad	1755.26	1833.76	5.15
Cosmos	5x200	TS_quad	123.52	532.14	1.84	HQA_quad	121.86	591.29	1.85

Table 12.6: The best results obtained for the larger problems of the Iridium 33 and Cosmos 2251 dataset in terms of energy and time with the new quadratization method.

Beyond the significant reductions in computational time, the new quadratization method also provides meaningful improvements in the energy values obtained by the solvers. Although the optimal solutions remain consistent between the two methods, the new approach's intrinsic quadratic formulation — avoiding the overhead introduced by auxiliary variables —dramatically reduces the problem's complexity. This streamlined approach allows solvers to explore a significantly less convoluted energy landscape, leading to better energy values and, consequently, higher-quality solutions.

These dual benefits — reduced computational charge and improved solution quality — are particularly critical in extreme scenarios. With the new quadratization method, cases involving 10 capture maneuvers with 114 debris objects or even 5 capture maneuvers with 200 debris objects become computationally tractable. In

Dataset	**C**	**D**	**Function Assembly Time (s)** **Old Method**	 **New Method**
Iridium	3	15	0.51	0.02
		20	1.34	0.04
		25	2.67	0.07
		50	44.18	0.21
		75	146.72	0.52
		100	188.59	0.95
		114	278.65	1.18
	4	25	321.30	0.08
		50	5949.25	0.31
	5	114	—	2.24
	10	114	—	4.61
Cosmos	5	200	—	6.90

Table 7: Assembly time with the old and the new quadratization method.

contrast, these scenarios would have been infeasible with the old method due to excessive function assembly times and high problem complexity.

By substantially reducing both computational overhead and problem difficulty, the new quadratization technique enables solvers to efficiently identify better solutions. This advancement provides a promising pathway for scaling ADR mission planning to larger and more complex datasets, facilitating the development of real-world applications in space debris management.

12.4 Discussion

The results of this work highlight the potential of quantum optimization techniques — in particular, QA and HQA — in addressing the complex combinatorial challenges associated with **multi-target ADR missions**. By formulating the ADR problem as a **QUBO** model, we proved that quantum solvers can efficiently explore the solution space, reducing computational complexity compared to classical metaheuristic methods such as SA, GA, and TS.

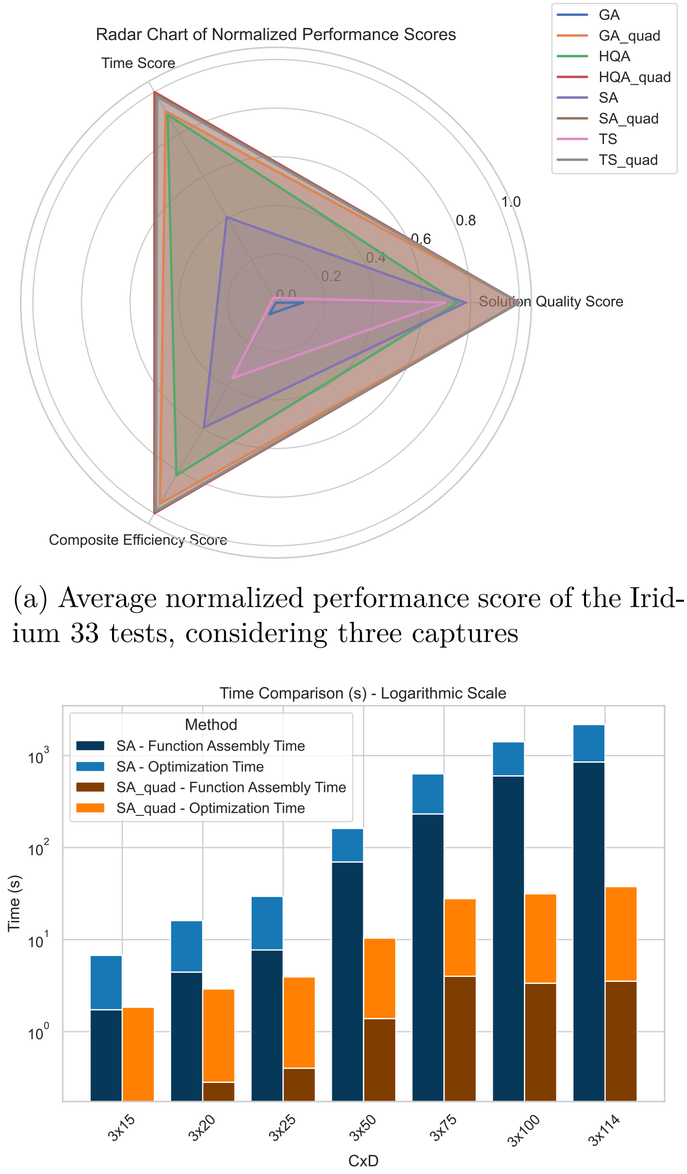
The key findings of this work are:

- **Computational Efficiency and Scalability:** Quantum-based methods, particularly HQA, maintained stable performance and outperformed classical solvers in terms of computational time for large-scale ADR scenarios. As problem size increased, classical methods exhibited **exponential growth in computational time**, whereas quantum solvers scaled more efficiently.
- **New Quadratization Method:** The new quadratization approach significantly **reduces function assembly times and problem complexity** by transforming high-degree polynomial constraints into a native quadratic form

without introducing auxiliary variables. This enables the application of QUBO-based solvers to larger ADR instances that were previously infeasible.

- **Constraint Handling Strategy:** Our modified constraint management approach for QA improves the probability of obtaining valid solutions while mitigating common issues related to energy landscape distortion.
- **Competitive Energy Solutions:** Across multiple test scenarios, quantum solvers lead to **near-optimal solutions**, proving their potential for real-world ADR mission planning.

These outcomes establish **quantum computing as a feasible tool for orbital debris management**, offering a pathway toward more efficient and scalable ADR mission planning.



(a) Average normalized performance score of the Iridium 33 tests, considering three captures

(b) Assembly and optimization time

Figure 12.4: Time, energies and scores considering 3 capture events collecting debris from Iridium 33 datasets with and without the new quadratization methods proposed.

Chapter 13

MATLAB/Simulink Mission Simulation

This chapter describes the MATLAB/Simulink simulation environment developed to validate the feasibility and accuracy of the debris capture sequences derived from the optimization. The simulations utilize MATLAB's Spacecraft Dynamics Blockset and Satellite Scenario toolkit to verify optimized mission solutions. Key mission metrics—particularly Time-of-Flight (TOF) and delta-v (Δv)—are compared against theoretical values obtained from the optimization process. The integration of advanced modeling and visualization tools enables comprehensive validation of the mission planning framework, bridging theoretical design with practical implementation.

13.1 Simulation Objectives

The MATLAB/Simulink simulation serves as a critical verification tool, assessing the practical feasibility of the capture sequences obtained through the ADR problem optimization. The primary objectives of the simulation are:

- **Validation of Optimized Solutions:** To ensure that the capture sequences proposed by the optimization are executable and feasible under realistic orbital mechanics scenarios.
- **Accuracy Assessment:** To quantitatively verify the correctness of the theoretical TOF and delta-v values, fundamental in the optimization formulation, by comparison with results obtained from detailed simulations.
- **Qualitative Visualization:** To utilize detailed visualizations, provided by MATLAB's Satellite Scenario toolkit, for qualitative assessments, ensuring intuitive verification of trajectory accuracy, maneuver correctness, and mission operational feasibility.

13.2 Simulation Environment

The simulation framework utilized a combination of advanced toolsets from MATLAB and Simulink to achieve accurate and realistic modeling of spacecraft trajectories:

- **Spacecraft Dynamics Blockset:** Used to model spacecraft maneuvers with precision [62]. In particular, it simulates Hohmann transfer maneuvers between debris orbits and the final disposal orbit. The model includes gravitational perturbations, such as Earth's J2 effect, ensuring accurate simulation of RAAN alignment and orbital transfers.
- **Satellite Scenario Toolkit:** The Satellite Scenario toolkit [63] offered capabilities for detailed mission visualization, allowing dynamic rendering and animation of spacecraft trajectories. This tool significantly aided in qualitative mission assessment and facilitated intuitive validation through visual confirmation.
- **Hohmann Transfer Modeling:** The Hohmann transfer approach follows the guidelines and methodologies described in MathWorks' official documentation [64], ensuring a physically accurate representation of transfer maneuvers.

13.3 Simulation Workflow

The simulation workflow involved the following structured steps:

1. **Sequence Input:** The optimal debris capture sequences identified by the optimization were input into the MATLAB environment as initial simulation conditions.
2. **Orbital Parameters and Initial Conditions:** Orbital data for debris and the chaser spacecraft are sourced from the dataset used in the optimization process, ensuring consistency with the theoretical analysis.
3. **Standardized Mission Durations:** For consistency, fixed durations of 2 hours for debris capture and 1 hour for release operations are applied across all simulations.
4. **Trajectory Simulation and Animation:** Each capture sequence simulation included the calculation of RAAN alignment using Earth's J2 perturbation, the execution of Hohmann transfer maneuvers, and a waiting time. The simulations were visually animated using MATLAB's Satellite Scenario toolkit, providing an additional qualitative validation step.

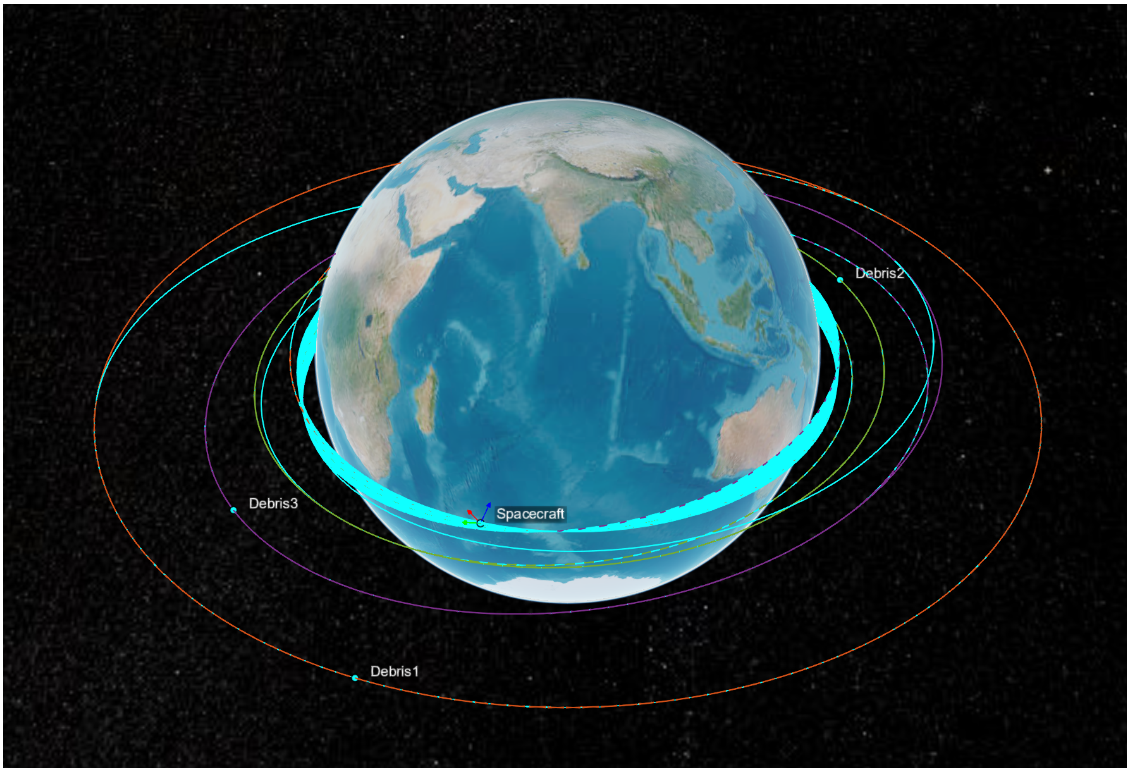


Figure 13.1: 3D visualization of the spacecraft and debris objects in Earth orbit, as generated by MATLAB's Satellite Scenario toolkit. The trajectories illustrate the chaser spacecraft's orbit (cyan) and the target debris objects (orange, green, and purple). The simulation provides qualitative validation of the optimized capture sequence and orbital maneuvers.

13.4 Results and Verification

Following each simulation, key mission performance metrics—namely, TOF and delta-v (Δv)—were collected. These values were rigorously compared to the theoretical calculations initially embedded in the optimization objective function. The results demonstrated strong agreement between the simulated and theoretical values, with minimal discrepancies within acceptable operational margins.

Specifically—as shown in Table 13.1— the verification confirmed:

- **TOF Validation:** Simulated TOF closely matched theoretical values from the optimization, indicating accurate modeling of orbital dynamics and realistic mission timelines.

- **Delta-v Accuracy:** The simulated delta-v required for maneuver execution is fully aligned with optimization-based predictions, thus validating fuel efficiency calculations integral to optimization criteria.

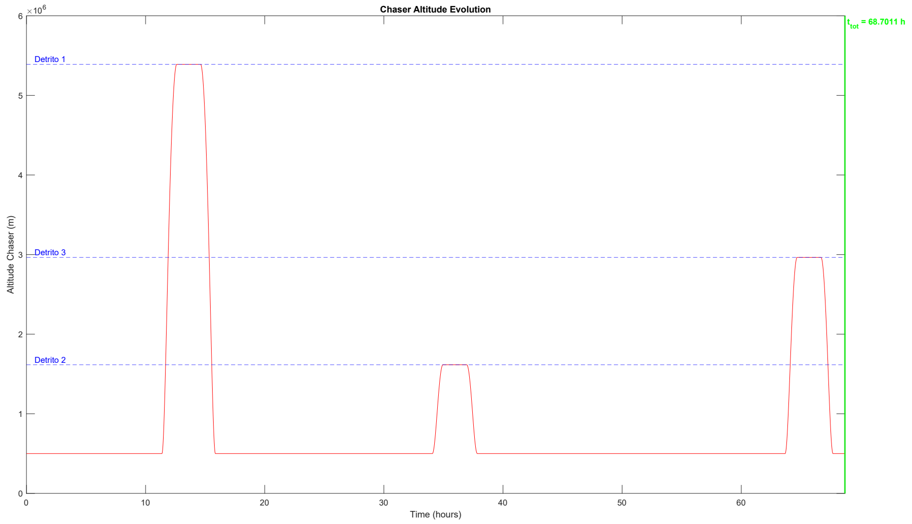


Figure 13.2: Time evolution of the chaser spacecraft's altitude over the entire mission duration. The plot highlights the different orbital altitudes corresponding to each debris rendezvous.

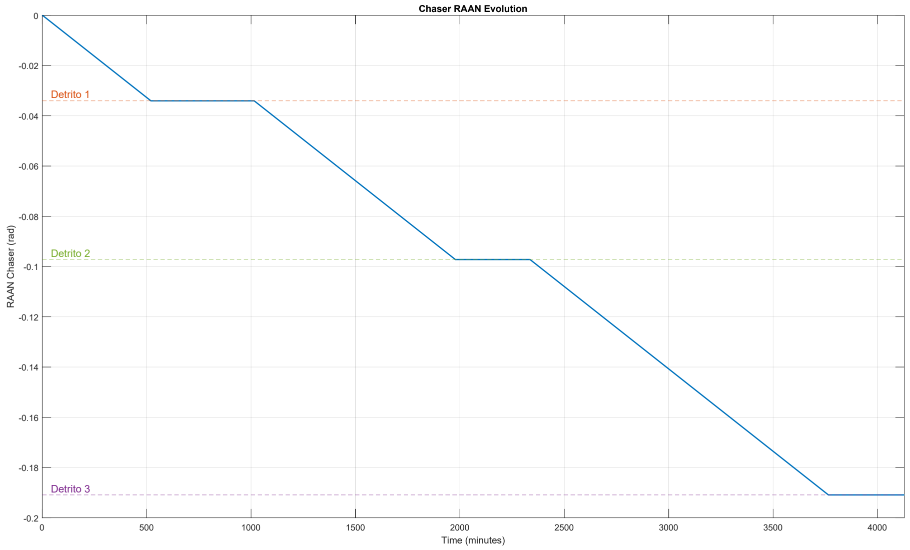


Figure 13.3: Evolution of the RAAN for the chaser spacecraft compared to the debris targets. The alignment points indicate optimal windows for transfer maneuvers, accounting for the precession due to Earth's J2 perturbation.

13.5 Discussion and Implications

The MATLAB/Simulink-based verification approach significantly strengthens the credibility and applicability of optimization solutions in practical aerospace mission planning. By demonstrating that simulated outcomes align closely with theoretical predictions, the proposed optimization framework's reliability and accuracy are

C	D	TOF (days)			Δv		
		Theoretical	**Simulated**	**Error**	**Theoretical**	**Simulated**	**Error**
2	15	673.11	673.46	0.05%	276.86	276.86	0.00%
	20	464.90	465.25	0.08%	276.86	276.86	0.00%
	25	632.67	633.02	0.06%	276.86	276.86	0.00%
	50	621.70	622.48	0.13%	257.37	257.37	0.00%
	75	674.91	674.91	0.00%	451.53	451.53	0.00%
	100	178.21	178.21	0.00%	451.53	451.53	0.00%
	114	52.63	52.62	0.02%	458.90	458.90	0.00%
3	15	508.76	509.12	0.07%	349.06	349.06	0.00%
	20	606.79	607.14	0.06%	349.06	349.06	0.00%
	25	162.26	162.59	0.20%	329.04	329.04	0.00%
	50	689.92	690.24	0.05%	503.71	503.71	0.00%
	75	569.61	569.73	0.02%	560.07	560.07	0.00%
	100	668.55	668.52	0.00%	404.46	404.46	0.00%
	114	555.40	555.39	0.00%	495.09	495.09	0.00%
4	25	1103.60	1103.77	0.02%	583.50	583.50	0.00%
	50	994.53	994.59	0.01%	375.99	375.99	0.00%

Table 13.1: Comparison between Theoretical and Simulated Results for TOF and Δv. Values from Iridium Dataset and HQA Solver.

substantially reinforced.

This validation highlights the practical potential of Quantum Annealing in solving complex aerospace mission planning challenges. Furthermore, it underscores the importance of simulation-based verification as a critical step in transitioning from theoretical optimization to practical implementation in space operations.

Part V

Conclusions

Chapter 14

Conclusions and Future Directions

This thesis has presented an innovative and comprehensive framework for optimizing multi-target Active Debris Removal (ADR) missions by leveraging emerging quantum computing paradigms, specifically Quantum Annealing (QA) and Hybrid Quantum Annealing (HQA). The methodology developed in this work addresses the pressing challenge of sustainable space operations, focusing on the removal of orbital debris—a growing threat to the security and operability of the near-Earth environment.

By reformulating the ADR trajectory optimization problem into a Quadratic Unconstrained Binary Optimization (QUBO) model, the research has demonstrated how quantum computing techniques can efficiently handle complex combinatorial optimization tasks. Traditional approaches to such problems, typically characterized by NP-hard complexity, often face limitations in scalability and computational time. In contrast, the application of QA and HQA shows tangible benefits in terms of computational speed, energy efficiency, and the quality of the solutions generated.

A distinctive contribution of this study lies in the development of an advanced quadratization technique. Unlike conventional methods, the proposed approach enables the transformation of higher-order polynomial constraints into quadratic expressions without the need for auxiliary variables. This advancement significantly reduces the overhead typically associated with embedding problems on quantum annealing hardware, which is often constrained by limited qubit connectivity and topology restrictions. The ability to directly quadratize complex objective functions enhances both the scalability and the practical feasibility of large-scale ADR mission planning.

The thesis also introduces an alternative strategy for constraint management, shifting the burden of constraint enforcement from the QUBO formulation phase to a dedicated post-processing stage. This methodological adjustment preserves the inherent advantages of quantum solvers—specifically their ability to explore vast solution spaces rapidly—while ensuring that mission-critical constraints are robustly satisfied. The approach leads to an overall improvement in the feasibility, quality, and robustness of the mission plans generated.

The comparative analyses conducted in this research have provided empirical evidence of the superiority of quantum optimization methods over classical meta-heuristic algorithms, including Simulated Annealing (SA), Tabu Search (TS), and Genetic Algorithms (GA). Quantum solvers demonstrated enhanced computational efficiency, reduced time complexity, and superior scalability, particularly when dealing with larger and more complex ADR scenarios. The integration of QA and HQA approaches into the ADR mission planning process represents a significant step toward more autonomous, efficient, and reliable mission design methodologies.

Despite these promising results, several limitations inherent to current quantum annealing hardware were identified. Among these are issues related to the limited number of available qubits, the constraints imposed by the hardware connectivity graph (often requiring complex minor embedding techniques), and the presence of thermal noise and decoherence, which can impact solution quality and repeatability. These challenges underscore the necessity for continued advancements in both quantum hardware and quantum algorithm design.

Looking ahead, future research should focus on several key areas to build upon the findings of this thesis:

1. **Hybrid Quantum-Classical Optimization:** The further development of hybrid algorithms that intelligently combine the strengths of quantum solvers with classical optimization techniques could yield significant improvements in solution quality, particularly for large-scale, real-world ADR missions.

2. **Quantum-Inspired Algorithms:** While quantum hardware is still maturing, quantum-inspired optimization algorithms running on classical architectures offer a promising intermediate step. Techniques such as Digital Annealing and Coherent Ising Machines could be explored as scalable alternatives for industrial applications.

3. **Error Mitigation and Robustness:** Addressing the challenges posed by noise and decoherence in current quantum annealers is critical. Advanced error mitigation techniques, along with more robust post-processing methods, will be essential to improve the consistency and reliability of quantum optimization outputs.

4. **Enhanced Quadratization Methods:** The quadratization method introduced in this work has shown promising results; however, further refinements could lead to additional efficiency gains. Exploring adaptive quadratization techniques that dynamically adjust based on the problem structure and the

target hardware characteristics represents an interesting avenue for future research.

5. **Application to Broader Space Mission Planning:** While this thesis focused on ADR mission planning, the methodologies developed have broader applicability. Future studies could explore their use in other space mission contexts, such as satellite constellation management, interplanetary trajectory design, and autonomous spacecraft operations.

In conclusion, this thesis has demonstrated the significant potential of quantum computing technologies, particularly Quantum Annealing and Hybrid Quantum Annealing, in addressing one of the most critical challenges of modern space exploration: the sustainable management of orbital debris. By proposing a scalable, efficient, and innovative framework for ADR mission optimization, this work contributes to advancing the state-of-the-art in both quantum optimization and space mission planning. It lays the groundwork for future research and development aimed at ensuring the long-term sustainability, safety, and accessibility of the space environment.

Continued investment in quantum technologies—both hardware and software—alongside international cooperation on debris removal initiatives, will be essential in transforming these theoretical advancements into practical solutions that safeguard humanity's shared orbital commons.

Bibliography

[1] Donald J Kessler, Nicholas L Johnson, JC Liou, and Mark Matney. The kessler syndrome: implications to future space operations. *Advances in the Astronautical Sciences*, 137(8):2010, 2010.

[2] Donald J. Kessler and Burton G. Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978.

[3] Nicholas L. Johnson, E. Stansbery, J.-C. Liou, M. Horstman, C. Stokely, and D. Whitlock. The characteristics and consequences of the break-up of the fengyun-1c spacecraft. *Acta Astronautica*, 63(1-4):128–135, 2008.

[4] Ting Wang. Analysis of debris from the collision of the cosmos 2251 and the iridium 33 satellites. *Science & Global Security*, 18(2):87–118, 2010.

[5] Nate Estell and collaborators. Space debris and iss. Technical report, National Aeronautics and Space Administration, 2023. Approved for Public Release via NASA STI Process DAA 20230010751.

[6] European Space Agency (ESA). Space debris by the numbers, 2024. Information last updated on 24 February 2025.

[7] Saket Kaurav Singh. Space debris – potential threat to space exploration. *Scientific Reports*, pages 20–24, 2021.

[8] Carmen Pardini and Luciano Anselmo. The short-term effects of the cosmos 1408 fragmentation on neighboring inhabited space stations and large constellations. *Acta Astronautica*, 210:465–473, 2023.

[9] European Space Agency (ESA). The kessler effect and how to stop it, 2024.

[10] Inter-agency Space Debris Coordination Committee, 2002. IADC Space Debris Mitigation Guidelines. UN COPUOS 40th session, Vienna.

[11] J.-C. Liou and Nicholas L. Johnson. A sensitivity study of the effectiveness of active debris removal in leo. *Acta Astronautica*, 64(2):236–243, 2009.

[12] J.-C. Liou, N.L. Johnson, and N.M. Hill. Controlling the growth of future leo debris populations with active debris removal. *Acta Astronautica*, 66(5):648–653, 2010.

[13] Hugh G. Lewis, Adam E. White, Richard Crowther, and Hedley Stokes. Synergy of debris mitigation and removal. *Acta Astronautica*, 81(1):62–68, 2012.

[14] Robin Biesbroek, Sarmad Aziz, Andrew Wolahan, Ste-fano Cipolla, Muriel Richard-Noca, and Luc Piguet. The clearspace-1 mission: Esa and clearspace team up to remove debris. In *Proc. 8th Eur. Conf. Sp. Debris*, pages 1–3, 2021.

[15] Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.

[16] Dario Izzo, Ingmar Getzner, Daniel Hennes, and Luís Felismino Simões. Evolving solutions to tsp variants for active space debris removal. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, page 1207–1214, New York, NY, USA, 2015. Association for Computing Machinery.

[17] Hong-Xin Shen, Tian-Jiao Zhang, Lorenzo Casalino, and Dario Pastrone. Optimization of active debris removal missions with multiple targets. *Journal of Spacecraft and Rockets*, 55(1):181–189, 2018.

[18] Lorenzo Federici, Alessandro Zavoli, and Guido Colasurdo. On the use of a* search for active debris removal mission planning. *Journal of Space Safety Engineering*, 8(3):245–255, 2021.

[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[20] Oliver Kramer. *Genetic Algorithms*, pages 11–19. Springer International Publishing, Cham, 2017.

[21] Jianan Yang, Xiaolei Hou, Yu Hen Hu, Yong Liu, and Quan Pan. A reinforcement learning scheme for active multi-debris removal mission planning with modified upper confidence bound tree search. *IEEE Access*, 8:108461–108473, 2020.

[22] Giulia Viavattene, Ellen Devereux, David Snelling, Niven Payne, Stephen Wokes, and Matteo Ceriotti. Design of multiple space debris removal missions using machine learning. *Acta Astronautica*, 193:277–286, 2022.

[23] Rupak Biswas, Zhang Jiang, Kostya Kechezhi, Sergey Knysh, Salvatore Mandrà, Bryan O'Gorman, Alejandro Perdomo-Ortiz, Andre Petukhov, John Realpe-Gómez, Eleanor Rieffel, Davide Venturelli, Fedir Vasko, and Zhihui

Wang. A nasa perspective on quantum computing: Opportunities and challenges. *Parallel Computing*, 64:81–98, 2017. High-End Computing for Next-Generation Scientific Discovery.

[24] Tobias Stollenwerk, Vincent Michaud, Elisabeth Lobe, Mathieu Picard, Achim Basermann, and Thierry Botter. Agile earth observation satellite scheduling with a quantum annealer. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5):3520–3528, 2021.

[25] Vinicius Marchioli, Mattia Boggio, Deborah Volpe, Luca Massotti, and Carlo Novara. Scheduling of satellite constellation operations in eo missions using quantum optimization. In *Optimization, Learning Algorithms and Applications*, pages 227–242, Cham, 2024. Springer Nature Switzerland.

[26] David Snelling, Ellen Devereux, Niven Payne, Matthew Nuckley, Giulia Viavattene, Matteo Ceriotti, Stephen Wokes, Giuseppe Di Mauro, and Harriet Brettle. Innovation in planning Space Debris removal missions using Artificial Intelligence and Quantum-inspired computing. In *8th European Conference on Space Debris*, 04 2021.

[27] Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki, Yasuhiro Watanabe, Kazuya Takemoto, and Hirotaka Tamura. Digital annealer for high-speed solving of combinatorial optimization problems and its applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 667–672, 2020.

[28] Thomas Swain. Optimisation of active space debris removal missions with multiple targets using quantum annealing. *arXiv preprint arXiv:2311.01852*, 2023.

[29] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

[30] Yan Wang, Jungin E. Kim, and Krishnan Suresh. Opportunities and challenges of quantum computing for engineering optimization. *Journal of Computing and Information Science in Engineering*, 23(6):060817, 08 2023.

[31] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J Egger, Bruce G Elmegreen, et al. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, pages 1–18, 2024.

[32] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

[33] Christos Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[34] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer Berlin, 01 2003.

[35] George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Ltd, 1988.

[36] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[37] D-wave systems. `https://www.dwavesys.com/`.

[38] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014.

[39] Maliheh Aramon, Gili Rosenberg, Elisabetta Valiante, Toshiyuki Miyazawa, Hirotaka Tamura, and Helmut Katzgraber. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics*, 7:48, 04 2019.

[40] I. G. Rosenberg. Reduction of bivalent maximization to the quadratic case. *CAH. CENTRE ET. RECH. OPERAT.; BELG.; DA. 1975; VOL. 17; NO 1; PP. 71-74; BIBL. 10 REF.*, 1975.

[41] Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225, 2002.

[42] Nike Dattani. Quadratization in discrete optimization and quantum mechanics, 2019.

[43] Neculai Andrei. *Penalty and Augmented Lagrangian Methods*, pages 185–201. Springer International Publishing, Cham, 2017.

[44] Yves Crama, Sourour Elloumi, Amélie Lambert, and Elisabeth Rodriguez-Heck. Quadratization and convexification in polynomial binary optimization. working paper or preprint, 2022.

[45] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.

[46] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass. : Addison-Wesley Pub. Co., 1989.

[47] Fred Glover. Tabu search - part i. *INFORMS Journal on Computing*, 2:4–32, 01 1990.

[48] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[49] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[50] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

[51] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[52] Satoshi Morita and Hidetoshi Nishimori. Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12):125210, 12 2008.

[53] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[54] Philipp Hauke, Helmut G Katzgraber, Wolfgang Lechner, Hidetoshi Nishimori, and William D Oliver. Perspectives of quantum annealing: Methods and implementations. *Reports on Progress in Physics*, 83(5):054401, 2020.

[55] Deborah Volpe, Giacomo Orlandi, and Giovanna Turvani. Improving the solving of optimization problems: A comprehensive review of quantum approaches. *Quantum Reports*, 7(1):3, 2025.

[56] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[57] Lynnane George. *Introduction to Orbital Mechanics*. 01 2023.

[58] David Vallado and Paul Crawford. SGP4 orbit determination. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.

[59] Philippe Codognet. Domain-wall / unary encoding in qubo for permutation problems. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 167–173, 2022.

[60] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[61] CelesTrak. Norad gp element sets.

[62] MathWorks. *Spacecraft Dynamics*. Available at: `https://it.mathworks.com/help/aeroblks/spacecraftdynamics.html`.

[63] MathWorks. *Satellite Scenario Overview*. Available at: `https://it.mathworks.com/help/aerotbx/ug/satellite-scenario-overview.html`.

[64] MathWorks. *Hohmann Transfer with the Spacecraft Dynamics Block*. Available at: `https://it.mathworks.com/help/aeroblks/hohmann-transfer-with-the-spacecraft-dynamics-block.html`.