

Mathematical Modeling and Optimization of Aerospace Maintenance Scheduling

Yasser Ghamary

A Thesis

in

The Department

of

Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Industrial Engineering) at

Concordia University

Montréal, Québec, Canada

September 2025

© Yasser Ghamary, 2025

**CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: *Yasser Ghamary*

Entitled: *Mathematical Modeling and Optimization of Aerospace Maintenance Scheduling*

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (*Industrial Engineering*)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Ryan O'Neil Chair

Dr. Guoqing Zhang External Examiner

Dr. Onur Kuzgunkaya Examiner

Dr. Masoumeh Kazemi Zanjani Examiner

Dr. Farnoosh Naderkhani Examiner

Dr. Mingyuan Chen Thesis Supervisor

Approved by _____
Dr. Ramin Sedaghati
Graduate Program Director

Sep 24, 2025 _____
Dr. Mourad Debbabi
Dean of Faculty of Engineering and Computer Science

Abstract

Mathematical Modeling and Optimization of Aerospace Maintenance Scheduling

Yasser Ghamary, Ph.D.
Concordia University, 2025

This thesis addresses the complex and computationally challenging problem of scheduling aircraft maintenance operations at the operational level. These maintenance activities are critical for ensuring aircraft safety and reliability but involve intricate interdependencies, resource constraints, and sequencing requirements that render the scheduling problem NP-hard.

To tackle this, we develop a mixed-integer linear programming (MILP) model that captures the key constraints of the problem, including task durations, inter-component dependencies, resource capacity limitations, and intra-task activity sequencing. Due to the computational intractability of solving the full MILP for real-world instances, we propose a hybrid solution approach combining a tailored Iterative Local Search (ILS) heuristic and a Lagrangian Relaxation (LR) framework.

The ILS heuristic is designed to efficiently generate high-quality feasible schedules by exploring the search space through domain-specific neighborhood moves and restart strategies. It serves as both a practical scheduling tool and a source of upper bounds for evaluating solution quality.

Complementing this, the LR framework decomposes the original problem by relaxing combinations of resource, dependency, and sequencing constraints. We employ a projected subgradient optimization method to solve the resulting dual problems, using adaptive step-size strategies to improve convergence stability. The LR approach provides lower bounds for the optimal solution, allowing us to assess the quality of heuristic solutions and gain insight into the impact of different constraint sets.

Comprehensive computational experiments on small and large problem instances demonstrate the effectiveness of the proposed methods. The ILS produces near-optimal solutions in reasonable time, while the LR method yields informative bounds and highlights structural properties of the

problem. Results show that adaptive step-size strategies enhance convergence behavior without increasing computational effort.

This thesis contributes a robust, scalable framework for operational maintenance scheduling and offers a foundation for future work in stochastic modeling, parallel computing, and real-time decision support for aviation maintenance operations.

Dedication

To my wife, Fatemeh —

You are my heart, my best friend, and my love. I couldn't have made it here without your incredible support. With you in my life, I found love, hope, dreams, and purpose. I dedicate this work to you above all, because it belongs to you as much as it does to me. I will love you forever.

To my parents, Hossein and Mehrangiz —

To my father, who has always been my role model — your work ethic, wisdom, and quiet strength have shaped my character and guided me through every phase of life. You taught me the value of discipline and purpose, and I am endlessly grateful.

To my mother, whose endless love and selflessness have been a constant source of comfort and motivation — your gentle heart, your prayers, and your quiet sacrifices have carried me further than you will ever know. You are the soul of our family, and I owe much of who I am to your care and presence.

To my supervisor, Dr. Chen —

Thank you for being the best supervisor I could have ever hoped for. Your guidance over the years has been invaluable — not only as a scholar but as a mentor and a friend. Your insight, encouragement, and thoughtful challenges helped shape this work and my growth as a researcher. You are, truly, a beautiful mind.

Acknowledgments

This research is supported in part by FRS-GCS funding of Concordia University, Canada, and NSERC discovery grant, Canada.

Table of Contents

List of Figures.....	x
List of Tables	xi
List of Algorithms	xii
1 Introduction.....	1
1.1 Background and Industry Growth.....	1
1.2 Overview of Maintenance Types	2
1.2.1 Planned vs. Unplanned Maintenance	2
1.2.2 Types of Planned Maintenance Checks	3
1.3 Importance of Scheduling and Labor Considerations	4
1.3.1 Criticality of Efficient Scheduling	4
1.3.2 Labor Constraints	5
1.3.3 Implications of Poor Scheduling	5
1.4 Economic Impact of Improper Maintenance and Delays	5
1.4.1 Costs of Delays	6
1.4.2 Aircraft-on-Ground (AOG) Costs	6
1.4.3 Airline Perspective on MRO Spending.....	6
1.5 Remanufacturing in Aerospace MRO.....	7
1.5.1 Definition and Process	7
1.5.2 Benefits for MRO.....	7
1.5.3 Challenges and Opportunities	8
1.6 Research Motivation and Problem Statement.....	8
1.6.1 Gap in Current Practices	8
1.6.2 Why New Models/Approaches Are Needed.....	9
1.7 Research Objectives and Questions	10
1.8 Contributions of This Thesis.....	11
1.8.1 Publications Resulting from This Thesis	12
1.9 Co-Authorship Statement.....	14

1.10	Thesis Organization	14
2	A Mathematical Model for Aerospace Product MRO Scheduling with Remanufacturing	16
2.1	Introduction.....	17
2.2	Literature Review.....	19
2.2.1	Advancements in Aircraft Heavy Maintenance Scheduling	19
2.2.2	Engine Maintenance Scheduling and Data-Driven Approaches.....	20
2.2.3	Remanufacturing in MRO: Emerging Perspectives.....	21
2.2.4	Broader Insights from Remanufacturing Literature	21
2.2.5	Gaps and the Way Forward.....	22
2.3	Problem Description and Modeling	22
2.3.1	Problem Description.....	22
2.3.2	Cost Analysis of Maintenance Operations	25
2.3.3	Project Cost Optimization Model.....	28
2.4	An MRO Scheduling Example with Different Scenarios	31
2.4.1	Data Sets and Scenarios	32
2.4.2	Operational Performance Analysis	38
2.4.3	Cost Analysis	48
2.4.4	Summary and Discussion.....	51
2.5	Conclusion	52
3	An Operational-Level MRO Scheduling Optimization for Aircraft Type C and D Checks Using Iterative Local Search	54
3.1	Introduction.....	55
3.2	Literature Review.....	56
3.2.1	Challenges in MRO Scheduling.....	57
3.2.2	Optimization Models in MRO Scheduling	57
3.2.3	Heuristic and Metaheuristic Methods in MRO Scheduling	58
3.2.4	Remanufacturing in Aviation MRO.....	59
3.2.5	Advanced Studies and Research Gap.....	60
3.2.6	Contribution of the Present Study	60
3.1.1	Authors' Contribution Table.....	61

3.3	Problem Description and Modeling	63
3.4	Solution Methodology	63
3.4.1	Constructive Heuristic.....	64
3.4.2	Local Search Heuristic	68
3.4.3	Iterative Local Search	72
3.5	Numerical Examples	76
3.5.1	Small-Sized Problem.....	77
3.5.2	Solutions of the Small-Sized Problem and Analysis.....	78
3.5.3	Computational Results of Larger-Sized Instances	83
3.5.4	Enhancing Computational Efficiency	85
3.5.5	Practical Implications for MRO Operations	86
3.6	Conclusion	86
4	Lagrangian Relaxation and Subgradient Optimizat-ion for Large-Scale MRO	
	Scheduling.....	89
4.1	Introduction.....	90
4.2	Literature Review.....	91
4.2.1	Historical Foundations and Core Concepts.....	91
4.2.2	Convergence and Algorithmic Enhancements	92
4.2.3	Expanding Applications and Sector-Specific Use Cases	92
4.2.4	Emerging Focus: Maintenance Scheduling and MRO.....	93
4.3	Problem Description and Constraint Structure	93
4.4	Lagrangian Relaxation Formulation	96
3.1.2	Relaxing Resource Constraints	96
4.4.1	Relaxing Component Dependency Constraints	98
3.1.3	Relaxing Activity Sequence Constraints.....	100
4.4.2	Relaxing Component Dependency & Resource Capacity Constraints	102
4.4.3	Relaxing Component Dependency & Activity Sequencing.....	104
4.4.4	Relaxing Resource Capacity & Activity Sequencing	106
4.4.5	Relaxing Resource Capacity, Component Dependency, & Activity Sequencing.....	107
4.5	Subgradient Optimization Framework.....	109
4.5.1	Initialization	110
4.5.2	Solving the Relaxed Subproblems	110

4.5.3	Subgradient Calculation	111
4.5.4	Subgradient Norm Computation	111
4.5.5	Multiplier Update	111
4.5.6	Step Size Selection	112
4.5.7	Convergence Criteria	112
4.5.8	Implementation Details	113
4.6	Numerical Results and Discussion.....	113
4.6.1	Test Instances.....	114
4.6.2	Relaxation Schemes	115
4.6.3	Results on Small Instances.....	115
4.6.4	Overall Interpretation and Conclusions	124
4.6.5	Gap Evolution and Stability – Adaptive Epsilon	125
4.6.6	Gap Evolution and Stability – Adaptive Step-Size Epsilon.....	130
4.6.7	Large-Scale Instance Testing: Scenario Selection and Rationale.....	133
4.6.8	Results on Large-Scale Instance	134
4.7	Conclusion	137
5	Conclusion and Future Work	139
5.1	Summary of Contributions.....	139
5.2	Recommendations for Future Work.....	140
5.3	Closing Remarks	141
Appendix A		142
Appendix B		143
Biography.....		149

List of Figures

Figure 2-1	Network of inter-dependencies among components	23
Figure 2-2	Relationship diagram for tasks and activities.....	24
Figure 2-3	Cost Terms and Comparison	49
Figure 4-1	Gap Progression Plot – Scenario A	118

Figure 4-2 Gap Progression Plot – Scenario B	119
Figure 4-3 Gap Progression Plot – Scenario C	120
Figure 4-4 Gap Progression Plot – Scenario D	121
Figure 4-5 Gap Progression Plot – Scenario E	122
Figure 4-6 Gap Progression Plot – Scenario F.....	123
Figure 4-7 Gap Progression Plot – Scenario G	124
Figure 4-8 Gap Progression Plot – Scenario D, Configuration 1	128
Figure 4-9 Gap Progression Plot – Scenario D, Configuration 2	128
Figure 4-10 Gap Progression Plot – Scenario D, Configuration 3	129
Figure 4-11 Gap Progression Plot – Scenario D, Adaptive Step-Size Epsilon.....	132
Figure 4-12 Gap Progression Plot – Scenario D, Large Instance – Constant Epsilon.....	136
Figure 4-13 Gap Progression Plot – Scenario D, Large Instance – Adaptive Epsilon	136

List of Tables

Table 2-1 Values for Component 2 in Scenario 1.....	33
Table 2-2 Parameters used in in HFD calculation in scenario 1	35
Table 2-3 Component information and unplanned maintenance cost	36
Table 2-4 Expected maintenance tasks in the next maintenance window	36
Table 2-5 Current and future maintenance costs	37
Table 2-6 Failure probability for each component depending on the task performed on it.....	37
Table 2-7 Results for Scenario 1 (and Scenario 4)	39
Table 2-8 Result for Scenario 2A (2 unit of resources)	42
Table 2-9 Table 2.9 Result for scenario 2B (8 unit of resources).....	43
Table 2-10 Result for Scenario 3	45
Table 2-11 Result for Scenario 5	47
Table 3-1 Comparison of Existing Studies and Present Work	61
Table 3-2 Optimal Solution From CPLEX	79
Table 3-3 Constructive Heuristic Solution	80
Table 3-4 Local Search Heuristic Solution	81
Table 3-5 Iterative Local Search (ILS) Solution.....	82

Table 3-6 Comparative Results and Computational Performance.....	82
Table 3-7 Results for Solution Methodology on Large-Sized Problem	84
Table 3-8 Results with Limited Neighborhood Size (6 Neighbors)	85
Table 4-1 Relaxation Schemes for Subgradient Optimization	116
Table 4-2 Summary of Results for Each Scheme	116
Table 4-3 Performance Comparison on Large Instance (Scenario D)	135

List of Algorithms

Algorithm 3-1 Constructive Heuristic for Initial Scheduling	65
Algorithm 3-2 Local Search Heuristic	69
Algorithm 3-3 Iterative Local Search (ILS) Heuristic	73

Chapter 1

Introduction

1.1 Background and Industry Growth

The aviation industry has experienced robust expansion over the past few decades, consistently growing at around 7% annually ([International Air Transport Association, 2019](#)) and with air passenger journeys projected to rise by an average of 4.7% per year ([Boeing, 2023](#)). This upward trajectory has led to a marked increase in the number of aircraft in service and, consequently, a corresponding surge in the demand for Maintenance, Repair, and Overhaul (MRO) operations ([International Air Transport Association, 2017](#)). The resulting maintenance workload is substantial, involving a broad range of scheduled checks (A, B, C, D) and urgent unscheduled interventions to keep aircraft airworthy and operational ([Pimapunsri & Weeranant, 2018](#)).

From an economic standpoint, airlines face considerable pressure to maintain profitability given that their operating margins can be as low as 1.8% in certain markets ([Tretheway & Markhvida, 2013](#)). Furthermore, maintenance costs alone can account for 9–13% of an airline’s operating budget, equating to hundreds of millions of dollars spent annually by major carriers ([International Air Transport Association, 2018](#)). These costs are driven by multiple factors, including labor wages for skilled technicians—estimated to constitute nearly half of overall MRO expenses—and expensive machinery, spare parts, and other resource requirements. Delays and cancellations tied directly to inadequate or untimely maintenance can add further financial strain, resulting in tens of millions of dollars in penalties and lost revenue each year ([Cook et al., 2012](#)).

At the same time, the aviation market’s growth is amplifying customer expectations for reliability and on-time performance. Overextended downtime due to maintenance activities or unforeseen failures not only erodes customer satisfaction but also imposes high Aircraft-on-Ground (AOG) costs that escalate with each hour the aircraft is out of service ([Stoller, 2010](#)). Recent industry investigations indicate that flight delays and cancellations attributable to maintenance deficits can reach or exceed \$20 million annually for large airlines ([Cook et al., 2004](#)). Given these mounting pressures, there is increasing recognition of how critical effective scheduling methods are in ensuring efficient utilization of both resources (labor, machines) and ground time.

The rapid advancement of aircraft technology—including the proliferation of sensors and enhanced data analytics—has further underscored the importance of robust maintenance systems. Many carriers are transitioning from traditional time-based maintenance toward condition-based or predictive maintenance strategies, aiming to detect potential failures before they occur (Verhagen & De Boer, 2018). By capturing real-time aircraft health data, airlines can plan maintenance events more accurately, reduce the risk of unplanned groundings, and ultimately lower overall MRO costs. However, these benefits hinge on strong scheduling practices that carefully align maintenance tasks with resource availability, flight timetables, and emerging remanufacturing solutions—an approach that not only keeps pace with industry growth but actively leverages it to achieve higher levels of operational efficiency.

1.2 Overview of Maintenance Types

Building on the growing importance of cost control and operational reliability discussed in the previous section, the broader Maintenance, Repair, and Overhaul (MRO) process typically falls into two main categories: planned and unplanned maintenance. Understanding these categories is key to devising effective strategies that keep aircraft in optimal condition while minimizing the financial and operational impacts of downtime.

1.2.1 Planned vs. Unplanned Maintenance

Planned maintenance encompasses the regularly scheduled inspections and tasks performed at predetermined intervals, guided by flight hours, flight cycles, or calendar dates (Kinnison & Siddiqui, 2013). The goal is to anticipate potential failures through systematic checks—commonly known as A, B, C, and D checks—and address them before they escalate into larger safety or cost issues (Kinnison & Siddiqui, 2013). By adhering to set schedules, airlines can better allocate resources (labor, equipment) and avoid disruptive last-minute interventions. These scheduled tasks may range from quick, routine checks to more in-depth procedures requiring weeks or months of aircraft downtime (Boere, 1977).

In contrast, unplanned (unscheduled) maintenance occurs when unexpected faults arise, whether due to mechanical breakdowns, pilot-reported anomalies, or unforeseen incidents (Kinnison & Siddiqui, 2013). This type of maintenance can prove particularly costly because it disrupts flight

schedules, potentially leading to flight cancellations, increased Aircraft-on-Ground (AOG) time, and knock-on effects throughout an airline's route network (Stoller, 2010). Even a single unplanned incident can ripple across multiple flights, incurring substantial financial and reputational risks. Consequently, while planned maintenance provides the backbone of a proactive safety and reliability strategy, effectively managing unplanned maintenance remains a core challenge in minimizing downtime and maintaining smooth operations.

1.2.2 Types of Planned Maintenance Checks

Building on the distinction between planned and unplanned maintenance, planned checks are further divided into A, B, C, and D categories, each defined by frequency, inspection depth, and resource demands (Kinnison & Siddiqui, 2013). These checks form a structured framework to ensure that aircraft remain airworthy and adhere to regulatory guidelines.

- **Type A and B Checks:** A-checks typically occur every two to three months, involving short yet systematic inspections of the aircraft's interior and exterior components (Kinnison & Siddiqui, 2013). Tasks might include visual checks, filter replacements, and minor lubrications, usually performed overnight or during brief ground stops to minimize flight disruptions. B-checks—though less universally recognized than A and C checks—are similar in spirit but can encompass a slightly wider range of tasks depending on the airline's specific procedures (Kinnison & Siddiqui, 2013). Both A and B checks generally demand fewer resources and entail minimal downtime compared to more extensive checks.
- **Type C Checks:** C-checks occur at longer intervals, usually between 18 and 24 months (Kinnison & Siddiqui, 2013). They represent a more in-depth inspection of the aircraft's structural and functional integrity, requiring parts of the fuselage, wings, and internal systems to be opened or disassembled for detailed examination (Kinnison & Siddiqui, 2013). Because of this expanded scope, C-checks often ground the aircraft for several days to weeks, making them more labor and resource-intensive than A or B checks. Nonetheless, they play a critical role in catching potential failures well ahead of operational disruptions.
- **Type D Checks:** Often regarded as the most extensive or “heavy” maintenance, D-checks take place approximately every six years, though the exact timetable can vary based on flight cycles and airline policies (Kinnison & Siddiqui, 2013). During a D-check, the

aircraft is stripped down extensively, with major structural components and systems removed, inspected, repaired, or overhauled (Kinnison & Siddiqui, 2013). This process can last weeks or even months, requiring highly skilled labor, specialized machinery, and significant logistical planning. While D-checks are costly and time-consuming, they are essential for uncovering any deep-seated structural or mechanical issues that could compromise long-term safety and reliability.

In summary, these planned checks form a tiered approach to aircraft maintenance, with A and B checks offering relatively quick, routine services, C-checks providing a more thorough mid-range inspection, and D-checks delivering the most comprehensive examination of the aircraft. By adhering to this structured schedule, airlines can mitigate the likelihood of unplanned failures and maintain a higher standard of operational safety.

1.3 Importance of Scheduling and Labor Considerations

Transitioning from the structured sequence of planned checks, the way these maintenance tasks are scheduled has a profound influence on aircraft availability, workforce utilization, and ultimately, overall safety (Samaranayake & Kiridena, 2012). Even with a well-defined maintenance framework (A, B, C, D checks), airlines often struggle to align required tasks with the real-world constraints of labor availability and resource capacity, which can escalate costs and operational risks if handled poorly (Weide et al., 2022).

1.3.1 Criticality of Efficient Scheduling

Effective scheduling serves as the cornerstone of maintenance operations, enabling airlines to minimize ground times while ensuring that safety-critical tasks are performed thoroughly (Deng et al., 2020). A single day of unplanned downtime can result in significant revenue losses, particularly when aircraft demand is high and flight schedules are tightly packed (Cook et al., 2004). In addition, short turnaround times for tasks such as A-checks or minor repairs prevent smaller issues from cascading into larger, more disruptive problems that would otherwise require substantial downtime in a busy hangar (Kinnison & Siddiqui, 2013). By systematically planning these intervals—especially for heavier checks like C and D—airlines can streamline labor usage, preventing technician overload in peak periods and reducing the likelihood of delays and errors.

1.3.2 Labor Constraints

Central to effective scheduling is the availability of skilled technicians and specialized machinery (Samaranayake & Kiridena, 2012). Aircraft mechanics, avionics specialists, and other highly trained personnel are indispensable for safe MRO operations, yet labor shortages in some regions are already driving up wages and complicating capacity planning (U.S. Bureau of Labor Statistics, n.d.). Coupled with specialized equipment—such as advanced testing rigs or large assembly tools—being limited or prohibitively expensive, these labor and machine constraints often become bottlenecks. If not properly accounted for in the scheduling process, these limitations can trigger extensive delays, inflate overhead costs, and compromise long-term fleet reliability (Weide et al., 2022).

1.3.3 Implications of Poor Scheduling

When maintenance tasks are scheduled sub-optimally or resource management is neglected, airlines can face ballooning costs tied to flight delays, last-minute crew reassignments, and penalties from failing to meet regulatory deadlines (Deng et al., 2020). Aircraft-on-Ground (AOG) scenarios, in particular, multiply these losses, as each idle aircraft can represent tens or hundreds of thousands of dollars of foregone revenue per day (Cook et al., 2004). Beyond direct financial repercussions, chronic scheduling inefficiencies erode workforce morale—overworked technicians may be prone to human error—and undermine passenger confidence when flight delays become frequent (Samaranayake & Kiridena, 2012). Thus, precise scheduling is not merely an administrative concern but a fundamental pillar that ensures both cost-effectiveness and operational excellence in an industry defined by narrow margins and high safety standards.

1.4 Economic Impact of Improper Maintenance and Delays

Even with structured scheduling strategies and sufficient labor resources, improper or delayed maintenance can trigger a cascade of financial setbacks for airlines. These setbacks include penalties imposed by regulators, lost revenue from flight cancellations, and additional costs linked to passenger compensation or rebooking (International Air Transport Association, 2017). Given that most commercial airlines operate on slim profit margins, unplanned expenses can have an outsized impact on annual earnings.

1.4.1 Costs of Delays

Flight delays attributed to maintenance deficits often create significant direct and indirect costs. One investigation found that in 2010 alone, 65,000 U.S. airline flights could not depart due to inadequate maintenance, incurring \$28.2 million in penalty fees across 25 airlines (Stoller, 2010). Additional expenses arise from re-routing passengers, covering their accommodations, or offering compensation for missed connections—factors that further inflate the total cost of maintenance-related disruptions (International Air Transport Association, 2017). Moreover, time spent on rescheduling and crew reassignments pulls operational staff away from routine planning and drives up overhead. Studies estimate that flight delays collectively cost the U.S. airline industry around \$7.7 billion in a single year, largely due to over 100 million minutes of delay linked to various factors, including unscheduled maintenance (Cook et al., 2004).

1.4.2 Aircraft-on-Ground (AOG) Costs

Whenever an aircraft remains grounded, it not only foregoes potential revenue but also can disrupt the larger network of flights. Aircraft-on-Ground (AOG) status is particularly costly because each out-of-service day can equate to lost fares, idle crews, and cascading schedule delays (Cook et al., 2004). Compounding the problem, airlines may have to lease replacement aircraft at high short-term rates or temporarily reduce service on high-demand routes. Thus, any deficiency in maintenance planning can rapidly multiply into substantial financial penalties, tarnished brand reputation, and disrupted customer relationships.

1.4.3 Airline Perspective on MRO Spending

From a broader standpoint, maintenance and repair costs can consume between 9% and 13% of an airline's operating budget, translating to hundreds of millions of dollars annually for larger carriers (International Air Transport Association, 2018). As air traffic demand continues to rise, these figures are only expected to increase. The synergy between effective scheduling (discussed in Section 1.3) and proactive maintenance programs becomes a strategic imperative for airlines seeking to control overhead, reduce unexpected downtime, and sustain profitability. By mitigating delays and avoiding substantial AOG intervals, well-planned maintenance can help maintain fleet

reliability while managing costs in an industry where even a small scheduling misalignment may lead to outsized repercussions.

1.5 Remanufacturing in Aerospace MRO

Building on the discussion of maintenance costs and scheduling complexities, remanufacturing has gained traction as an innovative strategy to further reduce expenses, manage part shortages, and improve overall MRO efficiency (Demirel & Gökçen, 2008). Rather than discarding or merely repairing worn-out components, remanufacturing restores these parts to a “like-new” or even superior condition by applying advanced refurbishment processes.

1.5.1 Definition and Process

Remanufacturing entails disassembly, thorough inspection, and replacement or reconditioning of subcomponents, followed by a reassembly that adheres to stringent quality standards (Kenné et al., 2012). In many cases, specialized machinery and processes such as precision machining, heat treatments, or specialized coating applications are used to bring the part’s performance back to original specifications (Huang et al., 2016). By reusing core assemblies and only substituting damaged or degraded elements, airlines can significantly extend the operational life of key components at a fraction of the cost compared to purchasing new ones (Demirel & Gökçen, 2008).

1.5.2 Benefits for MRO

From a cost perspective, remanufacturing can offer substantial savings by reducing the need for brand-new replacements—particularly for expensive components like engine modules, landing gear systems, or avionics (Kozik, 2016). This approach not only alleviates supply-chain bottlenecks but also shortens lead times, as refurbished parts can frequently be put back into service faster than acquiring new inventory (Zheng et al., 2019). Another increasingly important advantage is the environmental benefit. By preserving existing materials and minimizing waste, remanufacturing aligns well with corporate sustainability goals and circular economy principles (Huang et al., 2016). For MRO organizations, adopting remanufacturing translates to improved resource utilization and a potential competitive edge through cost-effective, environmentally responsible service offerings.

1.5.3 Challenges and Opportunities

Despite its promise, remanufacturing is not without barriers. First, technological hurdles arise due to the need for specialized equipment capable of diagnosing complex failure modes, as well as ensuring consistent quality in refurbished components (Hashemi et al., 2016). Second, logistical complications emerge when forecasting the arrival timing and condition of used parts. Aircraft wear can be influenced by diverse factors such as flight routes, environmental conditions, and operator practices (Kozik, 2016). Effective scheduling remains crucial to ensure remanufacturing tasks are aligned with overall MRO operations and do not inadvertently create bottlenecks. Nevertheless, these challenges also present a significant opportunity: as remanufacturing methods advance and airlines strive for more cost-effective and eco-friendly maintenance strategies, remanufacturing is poised to be a cornerstone of next-generation MRO solutions (Huang et al., 2016).

1.6 Research Motivation and Problem Statement

The rapid expansion of air travel and the growing complexity of Maintenance, Repair, and Overhaul (MRO) activities, as outlined in the previous sections, underscore a critical reality: existing practices often struggle to address the interconnected challenges of scheduling, resource allocation, and emerging remanufacturing opportunities (Samaranayake & Kiridena, 2012). While planned maintenance (A, B, C, and D checks) provides a structured framework for preventive actions, actual implementation can be hindered by labor shortages, constrained machine capacity, and unplanned failures that disrupt even the best-laid schedules (Pimapunsri & Weeranant, 2018). Moreover, as airlines pursue cost-reduction strategies in response to tight profit margins, there is growing pressure to integrate remanufacturing—a process promising substantial savings and environmental benefits—into the mainstream MRO pipeline (Huang et al., 2016).

1.6.1 Gap in Current Practices

Despite the recognized potential of remanufacturing to extend the life of costly components and mitigate supply-chain uncertainties, airlines face significant barriers in combining it with day-to-day scheduling processes. Traditional methods typically treat maintenance actions (repair, replacement) as isolated tasks, without fully considering how remanufacturing could optimize both

current and future costs (Demirel & Gökçen, 2008). Furthermore, conventional scheduling systems, which rely on manual planning or basic software tools, often neglect the dynamic nature of component risk—where a single unaddressed fault can lead to cascading failures and aircraft-on-ground (AOG) events (Pogačnik et al., 2017). This disconnect between long-term cost effectiveness (via remanufacturing) and short-term operational imperatives (e.g., meeting flight schedules) creates a gap in MRO practices that can inflate downtime and overall expenses (Samaranayake & Kiridena, 2012).

Another complicating factor is that maintenance tasks increasingly rely on specialized labor, making capacity planning an intricate balancing act (Pimapunsri & Weeranant, 2018). Airlines might have enough mechanics on paper to handle routine checks, but if a large proportion of those technicians are tied up in a lengthy D-check or an unexpected remanufacturing project, fewer resources remain to handle emergent issues. Current scheduling models generally fail to account for multi-task concurrency and do not adequately weigh the trade-offs between using technicians for immediate short-term fixes and dedicating them to deeper overhauls or remanufacturing tasks that can yield greater long-term benefits.

1.6.2 Why New Models/Approaches Are Needed

To address these shortcomings, the industry requires more robust and comprehensive scheduling solutions—ones that integrate cost analysis, remanufacturing decisions, resource constraints, and dynamic component reliability into a single framework. Beyond their operational intricacy, such scheduling problems are known to be NP-hard, implying that exact optimization can become computationally infeasible for large fleets or extensive planning horizons (Pimapunsri et al., 2022). This computational challenge is further magnified when incorporating risk mitigation strategies (e.g., factoring in the probability of component failures or the financial impact of unplanned events) and when modeling remanufacturing lead times alongside repair and replacement options.

In addition, the shift toward data-driven maintenance—where real-time sensor data informs predictive maintenance actions—demands more dynamic scheduling algorithms that can respond to updates in component health. Without advanced models, airlines risk adopting only a partial view of maintenance planning, focusing on immediate, visible costs rather than optimizing for the

full lifecycle of an aircraft part. This can lead to underutilization of remanufacturing, suboptimal resource use, and ongoing susceptibility to AOG incidents.

Hence, innovative approaches are essential not merely to handle the complexity of scheduling but also to enable strategic decision-making around risk, cost, and longevity of vital aircraft components. By bridging the gap between traditional scheduling frameworks and the next generation of maintenance strategies, new models can help airlines minimize downtime, leverage remanufacturing effectively, and ensure that safety and reliability remain paramount even in an environment of ever-evolving industry pressures.

1.7 Research Objectives and Questions

Drawing together the preceding discussions on maintenance complexities, remanufacturing integration, and scheduling constraints, the core aim of this research is to develop and evaluate a comprehensive MRO scheduling framework that incorporates cost optimization, risk mitigation, and remanufacturing opportunities. By addressing the identified gaps—particularly the fragmented consideration of maintenance tasks and the lack of robust methods to account for resource capacity, future maintenance needs, and real-time component risks—this study aspires to help airlines minimize downtime, reduce total operational costs, and maintain high reliability in an evolving industry landscape.

Building upon this overarching goal, the specific objectives of the research can be summarized as follows:

1. **Formulate an Integrated MRO Scheduling Model:** Develop a decision-making framework that simultaneously considers planned maintenance intervals, unplanned failure risks, and remanufacturing options. The model should explicitly capture labor and machinery constraints, along with potential penalties or bonuses for project delays and early completions.
2. **Incorporate Remanufacturing Strategies:** Investigate how remanufacturing can be systematically included in scheduling decisions to extend component life, reduce spare-part bottlenecks, and balance short-term repair activities with longer-term cost benefits.

3. **Propose and Validate Efficient Solution Approaches:** Given the NP-hard nature of scheduling problems in large-scale airline operations, design and test heuristics, metaheuristics, or other computationally viable strategies capable of delivering high-quality solutions within a practical time frame.

Based on these objectives, the research aims to address the following key questions:

1. How can maintenance scheduling models be enhanced to integrate both planned and unplanned maintenance actions, along with remanufacturing decisions that minimize total MRO costs?
2. In what ways can the model account for the dynamic risk of component failure, ensuring that the scheduling process remains robust against uncertain events or emerging maintenance demands?
3. What computational methods are most effective for tackling large-scale scheduling under limited technician, machine, and time constraints, and how do they perform against traditional scheduling approaches?
4. How can remanufacturing be optimized within the overall maintenance workflow to reduce downtime, tackle part shortages, and align with evolving sustainability objectives in the aviation sector?

By answering these questions, the study seeks to provide airlines with actionable insights on how to unify day-to-day operational concerns (e.g., turnaround times, labor shifts) with longer-term strategic goals (e.g., cost reduction, asset longevity) through improved scheduling and maintenance practices.

1.8 Contributions of This Thesis

This thesis presents three main contributions that differentiate our work from the existing literature on aircraft Maintenance, Repair, and Overhaul (MRO) scheduling:

1. Integrated MRO Scheduling with Remanufacturing Decisions

A mathematical optimization model is developed to jointly consider repair, replacement, and remanufacturing actions for aircraft components. Unlike traditional MRO frameworks

that treat these decisions in isolation, the proposed model explicitly incorporates the economic and reliability trade-offs among maintenance strategies. This integration enables a more holistic planning approach that supports long-term asset health, cost efficiency, and risk-aware scheduling, particularly under supply chain and component availability constraints.

2. Efficient Heuristic for Operational-Level MRO Scheduling

To address the combinatorial complexity of real-world scheduling in Type C and D checks, we design a tailored Iterative Local Search (ILS) heuristic. This method captures essential operational features, including resource limitations (e.g., labor, machines), unplanned maintenance events, task precedence, and incentive structures. The heuristic consistently generates high-quality solutions in computationally feasible timeframes, making it well-suited for deployment in dynamic airline maintenance environments where exact optimization is often intractable.

3. Lagrangian Relaxation Framework for Lower Bound Analysis

Building on the operational-level model, we formulate and implement a Lagrangian relaxation framework to compute lower bounds on the scheduling problem. Several constraint relaxation scenarios are explored, and a subgradient optimization algorithm—enhanced with both fixed and adaptive step-size strategies—is employed to solve the dual problem. The framework is used to assess the quality of heuristic solutions, understand the impact of different constraint groups, and analyze convergence behavior. Experiments on small and large-scale instances provide empirical validation of the method’s scalability and bounding strength.

1.8.1 Publications Resulting from This Thesis

The research conducted in this thesis has led to the development of two core manuscripts and a third ongoing study, each corresponding to one of the major methodological contributions discussed throughout the document:

(1) A Mathematical Model for Aerospace Product MRO Scheduling with Remanufacturing

Y. Ghamary, M. Chen

Department of Mechanical, Industrial & Aerospace Engineering, Concordia University,
Montreal, Canada.

(Published in Journal of Remanufacturing)

- Introduce a novel MRO scheduling model that integrates repair, replacement, and remanufacturing decisions.
- Incorporate risk assessment into the cost structure for unplanned maintenance, future maintenance windows, and delays.
- Develop a linear (integer) programming formulation and demonstrate its application on small to medium-scale problems.
- Assess the impact of varying parameters (e.g., unplanned maintenance costs, remanufacturing lead times, available labor) on total project cost and operational risk through scenario-based analysis.

(2) An Operational-Level MRO Scheduling Optimization for Aircraft Type C and D Checks Using Iterative Local Search

Y. Ghamary, M. Chen

Department of Mechanical, Industrial & Aerospace Engineering, Concordia University,
Montreal, Canada.

(Submitted to Journal of Quality in Maintenance Engineering)

- Propose an Iterative Local Search (ILS) heuristic designed to handle large-scale operational scheduling, explicitly targeting Type C and D checks.
- Address practical constraints, including shared labor, machine capacity, penalty/bonus schemes, and the probability of component failures.
- Demonstrate that the ILS heuristic can achieve near-optimal solutions rapidly, outperforming naive or manual scheduling methods.
- Compare exact solutions from a commercial solver with results from heuristic approaches to validate model feasibility and effectiveness.

(3) A Lagrangian Relaxation and Subgradient Optimization Framework for Lower Bounding Aircraft MRO Schedules with Remanufacturing

Y. Ghamary, M. Chen

Department of Mechanical, Industrial & Aerospace Engineering, Concordia University,
Montreal, Canada.

- Develops a Lagrangian relaxation approach for deriving lower bounds on complex MRO scheduling problems.
- Investigates the impact of relaxing resource, dependency, and sequencing constraints across multiple scenarios.
- Employs subgradient optimization with both fixed and adaptive step-size control, evaluating stability and convergence.
- Applies the method to small and large-scale MRO instances, offering insights into solution quality and computational efficiency.

Together, these contributions and publications provide a comprehensive framework for MRO decision-making that moves beyond conventional scheduling approaches, unifying remanufacturing with dynamic, risk-informed scheduling to improve airline profitability and reliability in an era of increasing operational demands.

1.9 Co-Authorship Statement

I, Yasser Ghamary, am the principal author of all manuscript chapters included in this dissertation. However, these manuscripts were co-authored by my supervisor, Dr. Mingyuan Chen, whose insights and guidance significantly shaped the development of the underlying ideas, the modeling and solution approaches, and the interpretation of computational results. Dr. Chen also contributed to manuscript drafting and revision throughout the research process.

1.10 Thesis Organization

Building on the motivations, objectives, and contributions set out in this introductory chapter, the thesis is organized as follows:

- **Chapter 2** of this dissertation is derived from Paper 1 (“A Mathematical Model for Aerospace Product MRO Scheduling with Remanufacturing”), co-authored with Dr. Chen. This paper is published in Journal of Remanufacturing.
- **Chapter 3** contains the main results from Paper 2 (“An Operational-Level MRO Scheduling Optimization for Aircraft Type C and D Checks Using Iterative Local Search”), co-authored with Dr. Chen. Relevant findings were submitted for publication in the Journal of Quality in Maintenance Engineering.
- **Chapter 4** includes new, unpublished contributions focused on Lagrangian relaxation techniques for MRO scheduling. This chapter presents original methodological enhancements such as adaptive epsilon adjustment strategies, along with extensive convergence and scalability analysis. These results were developed independently as an extension of the previous works and have not yet been submitted for publication, though future dissemination in journals is planned.
- **Chapter 5** concludes the dissertation and outlines directions for future research, building on the modeling, heuristic, and decomposition techniques developed across the earlier chapters.

Throughout these chapters, the collaborative efforts between my supervisor and me have strengthened the methodological rigor and broadened the practical applicability of our research in aerospace MRO scheduling and remanufacturing optimization.

Chapter 2

A Mathematical Model for Aerospace Product MRO Scheduling with Remanufacturing

Building on the broader motivations presented in Chapter 1, this chapter delivers a linear programming model specifically designed for scheduling aircraft C-check and D-check maintenance with multiple component treatment options—replacement, repair, and remanufacturing. Developed within the high-stakes context of the aerospace industry, where MRO operations underpin global transportation and high-tech growth, the model explicitly incorporates risk of component failure prior to the next maintenance cycle. By doing so, it strives to minimize total maintenance cost while upholding strict safety protocols. A key feature of the proposed model is the allocation of maintenance resources—including labor teams and specialized machinery—whose finite capacities impose practical constraints on scheduling. Through scenario analyses, the chapter evaluates how shifts in resource availability, the cost of unplanned maintenance, and penalty or bonus schemes affect both the schedule and overall expenses. Significantly, we investigate how remanufacturing, as an alternative to repair or replacement, can reduce long-term costs under various conditions. The main contribution here is the novel integration of risk assessment into a resource-optimized MRO scheduling framework. By combining repair, replacement, and remanufacturing strategies with dynamic failure probabilities, the model offers a more comprehensive lens through which airlines can plan maintenance and allocate resources. Thus, this chapter provides both a theoretical foundation and practical guidelines to strengthen decision-making in MRO operations—particularly vital for heavy checks like C and D, where downtime costs and safety requirements converge.

***Bibliographical note.** The content used for this chapter is from [\(Ghamary & Chen, 2024\)](#)*

2.1 Introduction

Global air transportation is undergoing rapid growth, with an annual growth rate of 7%. Projections indicate air passenger journeys will grow by an average of 3.7% per year over the next two decades (International Air Transport Association, 2017). This growth has led to an increase in the number of aircraft in operation, which in turn, raises the demand for aircraft maintenance business. Rigorous maintenance ensures industry standards are met with passenger and equipment safety guaranteed. Adherence to stringent safety protocols is crucial for the industry's enduring success (Lagos et al., 2020).

Maintenance, Repair, and Overhaul (MRO) operations are critical in achieving passenger and aircraft safety. MRO operations are complex demanding extensive resources. They require a variety of maintenance tasks, each with specific needs such as skilled labor and specialized machinery (Kinnison & Siddiqui, 2013). MRO tasks generally fall into two categories: unscheduled maintenance due to unexpected malfunctions and scheduled maintenance mandated by regulatory intervals (Pimapunsri & Weeranant, 2018; Saraçyakupoğlu, 2021). Heavy maintenance, known as Type C and D checks, is crucial. These checks involve taking aircraft out of service for extensive periods, thus demanding careful planning.

The consequences of overlooking scheduled maintenance are severe. They include operational disruptions such as delays and cancellations, and in the worst cases, accidents (Stoller, 2010). Financial repercussions are also significant. Major airlines may face over \$20 million in annual costs due to maintenance-related delays (Cook et al., 2004). This represents a substantial part of airline operating budget, as much as 9% (Bureau of Transportation Statistics, 2017). Therefore, effective MRO planning and scheduling are essential to minimize airline dispatch delays, maintain safety standards, and ensure cost-efficient operations.

In MRO processes, each aircraft is treated as a unique “Project” with specified “Work Package (WP)” that encompasses the maintenance “tasks” to be addressed. A WP is developed based on the aircraft age, flight history and other operational factors to identify components due for maintenance. The WP does not predetermine the specific maintenance tasks but flags components for inspection (Pogačnik et al., 2017). This strategy allows maintenance plans to remain adaptable,

as the actual condition and necessary maintenance actions for the components are determined only after a thorough inspection. The results of the inspection direct the maintenance team on the subsequent steps required for each component such as repair, replacement, remanufacturing or “no action”.

Timely availability of spare parts is critical for smooth MRO operations while intermittent demand for such parts poses a logistical challenge. Delays in obtaining necessary spare parts can prolong maintenance activities leading to increased operational costs and potential penalties. Rushing orders to cover shortages can disrupt the supply chain, significantly raising overall project costs (Kozik, 2016; Pogačnik et al., 2017). For instance, the long lead time for procuring expensive parts for aircraft engine MRO projects can substantially increase operational expenses (Kozik, 2016). “Remanufacturing” is an effective strategy in response to these logistical issues. It involves refurbishing defective parts to their original condition. Favored for its cost benefits and alignment with sustainability efforts, remanufacturing is becoming a more popular choice (Samaranayake & Kiridena, 2012). With advances in parts reuse technology, the demand for remanufactured aircraft parts is expected to climb (Huang et al., 2016).

This chapter introduces a mathematical model designed for optimal MRO scheduling, explicitly featuring remanufacturing as one of the potential actions. The model’s central objective is to identify the optimal sequence of maintenance tasks (repair, replacement, remanufacturing, or no action) to minimize total costs while maintaining safety. This work stands out as an early application of mathematical modeling that integrates remanufacturing decisions with aerospace MRO scheduling considerations.

To situate this research within existing scholarship, Section 2.2 offers a literature review, highlighting relevant studies on scheduling, resource constraints, and remanufacturing in aerospace MRO. Section 2.3 describes the problem setting and presents the proposed mathematical model. A numerical example and scenario analyses are introduced in Section 2.4, providing insight into how different resource availabilities, cost parameters, and remanufacturing lead times can affect maintenance decisions. Finally, Section 2.5 presents the conclusions of this study and proposes directions for future research.

2.2 Literature Review

Research on aircraft MRO scheduling has evolved substantially over the past few decades, particularly in the realm of heavy maintenance. As early as the 1970s, maintenance scheduling depended greatly on the expertise of seasoned maintenance planners and was frequently time-intensive, sometimes consuming weeks to produce a workable plan (Boere, 1977). A notable leap forward during this era was Air Canada's maintenance operations simulation model (AMOS), which compressed planning time from weeks to mere hours and demonstrated that technology-driven approaches could streamline maintenance (Boere, 1977). This momentum has continued with the development of more advanced tools, such as IFS Maintenix (IFS, 2019), further enhancing scheduling efficiency. However, challenges persist, including manual parameter inputs and the difficulty of managing uncertainties in maintenance tasks.

2.2.1 Advancements in Aircraft Heavy Maintenance Scheduling

A number of robust methodologies have been proposed in the literature to overcome these persistent scheduling hurdles. (Samaranayake & Kiridena, 2012) and (Samaranayake et al., 2002) introduced an integrated framework for maintenance planning, emphasizing the coordination of operational needs with scheduling constraints to reduce inefficiencies. (Weide et al., 2022) leveraged a genetic algorithm (GA) to craft resilient schedules for heavy maintenance checks, particularly under uncertain conditions. Their approach notably minimized workload fluctuations and reduced the need for last-minute schedule revisions. Likewise, (Yan et al., 2008) proposed a binary integer programming model to generate optimal scheduling solutions under varying constraints, while (Deng et al., 2020) employed dynamic programming to tackle similar problems. These models accounted for diverse aspects such as repair costs and aircraft health management, illustrating the mounting complexity in MRO scheduling as airlines seek to cut costs and enhance reliability. Further contributions by (Qin, Ma, et al., 2020) and (Pimapunsri et al., 2022) addressed specialized facets—specifically, procurement challenges and resource-constrained scheduling in heavy maintenance. The refinement of these frameworks continued with (Semaan & Yehia, 2019), who incorporated stochastic elements for military aircraft maintenance via Monte Carlo simulations, offering improved decision-making tools. (Qin et al., 2019) extended scheduling complexity to hangar management, using mixed-integer linear programming (MILP) to align

maintenance plans with parking capacity and movement constraints. The model delivered scalable solutions, making it practical for larger airline operations. On a related note, (Sriram & Haghani, 2003) tackled maintenance scheduling and re-assignment, focusing on minimizing both maintenance and reassignment costs; they introduced a heuristic approach tailored to mid-sized airlines, showcasing how simplified solution methods can address real-world limitations in computation and data availability.

2.2.2 Engine Maintenance Scheduling and Data-Driven Approaches

While the research above centers on heavy checks (C and D), aircraft engine MRO scheduling poses its own set of unique challenges. (Reményi & Staudacher, 2014) demonstrated a simulation-based technique for identifying effective scheduling rules in engine maintenance, successfully coping with fluctuating demands and workloads. Their findings highlighted that decentralized scheduling strategies can sustain on-time delivery, even under high workload pressures. (Kennet, 1994) similarly explored engine maintenance from a behavioral perspective, analyzing how airline personnel make day-to-day scheduling decisions. Other works, such as (Antonakis & Giannakoglou, 2018) and (Stranjak et al., 2008), applied advanced simulation or agent-based methods to forecast and schedule engine overhauls under part shortages and other dynamic factors, reinforcing the need for adaptive and predictive capabilities in modern MRO. These studies confirm that while engine and airframe maintenance share overarching goals—safety, cost-efficiency—they often entail differing resource allocations, component lead times, and failure profiles. In parallel, data-driven methodologies have become increasingly prevalent for MRO planning and scheduling. (Y. Hu et al., 2021) proposed analytics-based maintenance techniques, suggesting that historical data can effectively guide scheduling decisions. (Verhagen & De Boer, 2018) zeroed in on unscheduled maintenance, applying historical data analysis to streamline last-minute repairs and reduce operational disruptions. (Kozik & Sęp, 2012) delved into artificial neural networks (ANN) for forecasting aircraft engine overhaul demands, illustrating the push toward machine learning and predictive analytics to anticipate maintenance requirements more accurately.

2.2.3 Remanufacturing in MRO: Emerging Perspectives

Beyond the scheduling realm, remanufacturing has garnered attention as a sustainable and cost-effective strategy for aerospace MRO. (Rajamani et al., 2004) conducted a cost-benefit analysis in condition-based engine maintenance, showing the potential savings when engines are remanufactured instead of replaced. (Yang & Zheng, 2016) and (Johnson et al., 1999) delved into critical remanufacturing processes within aerospace engine operations, emphasizing how reusing or refurbishing parts can significantly curtail material waste and reduce lead times. Likewise, (Asmatulu et al., 2013) emphasized the environmental advantages, underlining the industry's growing emphasis on green initiatives and circular economy principles. Further expanding the conversation, (Zheng et al., 2019) identified production planning and control challenges in aircraft parts remanufacturing, specifically highlighting unpredictable arrival of used parts and dynamic demand for reconditioned components. (Lee et al., 2001) underscored the information-driven facet of remanufacturing, using an aircraft fuel pump as a case study to illustrate the complexities in data collection, quality assurance, and logistics when refurbishing parts to a "like-new" standard.

2.2.4 Broader Insights from Remanufacturing Literature

Outside the aerospace domain, remanufacturing has a long track record of success in industries such as automotive and power generation. For example, (Macedo et al., 2016), (Vercraene et al., 2014), and (Zhang et al., 2011) each presented numerical models for hybrid manufacturing–remanufacturing systems, demonstrating how inventory management, production scheduling, and reverse logistics can be optimized jointly. (Kenné et al., 2012) contributed by analyzing random machine failures and their effect on production planning, a crucial factor in any remanufacturing system facing fluctuating operational conditions. Meanwhile, (Van Loon & Van Wassenhove, 2018) devised methodologies to gauge the economic and ecological value of remanufacturing, helping suppliers and manufacturers balance profitability with sustainability. Moreover, (Wakiru et al., 2018) used discrete event simulation to assess how various remanufacturing and maintenance strategies influence power plant availability, particularly for remote or less-serviced regions. (Y. Wang et al., 2020) concentrated on the reconditioning of used parts featuring multiple failure modes, proposing sophisticated models for determining optimal reconditioning schemes. (Ma et al., 2011) addressed disassembly planning for end-of-life recovery, emphasizing the

sometimes-intricate decision-making involved in cost–profit trade-offs and environmental repercussions.

2.2.5 Gaps and the Way Forward

From the above discourse, it is evident that aerospace MRO scheduling has undergone significant refinements over time, yet explicit considerations of remanufacturing within operational-level planning remain relatively sparse. While many studies aim to optimize timing of checks or address resource scheduling, they often do not fully integrate defective part management or dynamic decisions around reuse options. The engine-focused studies highlight certain facets of reconditioning or repair, but few extend these insights to holistic scheduling frameworks that weigh replacement, repair, and remanufacturing against future cost implications. Addressing this deficiency, the model presented in this chapter seeks to bridge these gaps by incorporating remanufacturing as a viable operational strategy alongside established scheduling methodologies. By considering labor constraints, penalties for delayed tasks, risk of component failure, and remanufacturing lead times, our approach offers a broader and more adaptive platform for decision-making in aerospace maintenance. Hence, the model introduced here stands to contribute both to the theoretical body of scheduling literature and to practical MRO management, enhancing safety and cost efficiency in a competitive, rapidly expanding industry.

2.3 Problem Description and Modeling

Aerospace MRO operations are critical for ensuring the safety and reliability of aircraft. This section presents a detailed description of the MRO scheduling problem and introduces a mathematical model to find optimal solutions for scheduling such complex processes of operations. The model is developed with the focus on the most comprehensive maintenance procedures, the C-checks and D-checks.

2.3.1 Problem Description

Scheduled maintenance for aircraft, such as C-checks and D-checks mandated by industry standards, occurs at predefined intervals, with C-checks typically every 18 to 24 months and D-checks approximately every six years (Deng et al., 2021). The MRO process is project-based, treating each aircraft as a new project with unique maintenance needs. The interval between these

scheduled maintenances is fixed; however, unexpected incidents can necessitate immediate, unscheduled maintenance. This unscheduled maintenance, often due to the failure of components which were not adequately maintained or inspected, may lead to urgent maintenance requirements, disrupting the airline's operation and causing flight cancellations.

Upon the aircraft's scheduled arrival for maintenance, a preliminary Work Package (WP) is prepared. This WP lists the components that are anticipated to require maintenance based on historical data and the aircraft's operational profile. However, the actual condition and status of these components are not verified until a detailed inspection is done. The diagram in Figure 2.1 depicts an example network of interdependencies among aircraft components within a small WP. These interdependencies are critical as they dictate the sequence of maintenance operations, which is often determined by the physical layout and accessibility of the components within the aircraft. for instance, access to certain components may be dependent on the removal of others, dictating a logical sequence for maintenance tasks.

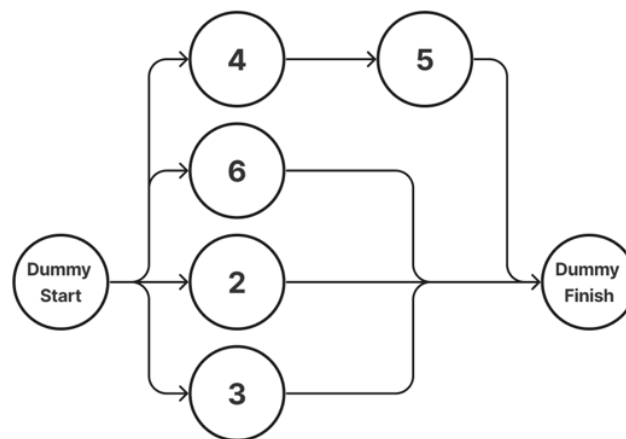


Figure 2.1: Network of inter-dependencies among components

The inspection reveals the conditions of each component and provides the necessary information to make informed decisions about the maintenance actions. Based on the inspection results, the project manager must decide the best course of actions for each component: whether to repair, replace, or remanufacture. These decisions significantly impact the maintenance workflow, dictating the number and order of the subsequent activities required.

For example, the decision to repair a component triggers a sequence of specific activities that must follow a predetermined order to ensure the integrity of the repair task. Interrelated activities in

each maintenance task are illustrated in the diagram in Figure 2.2. The diagram shows the importance of deciding the appropriate maintenance task: replacement, repair, remanufacturing, or in some cases, taking no action.

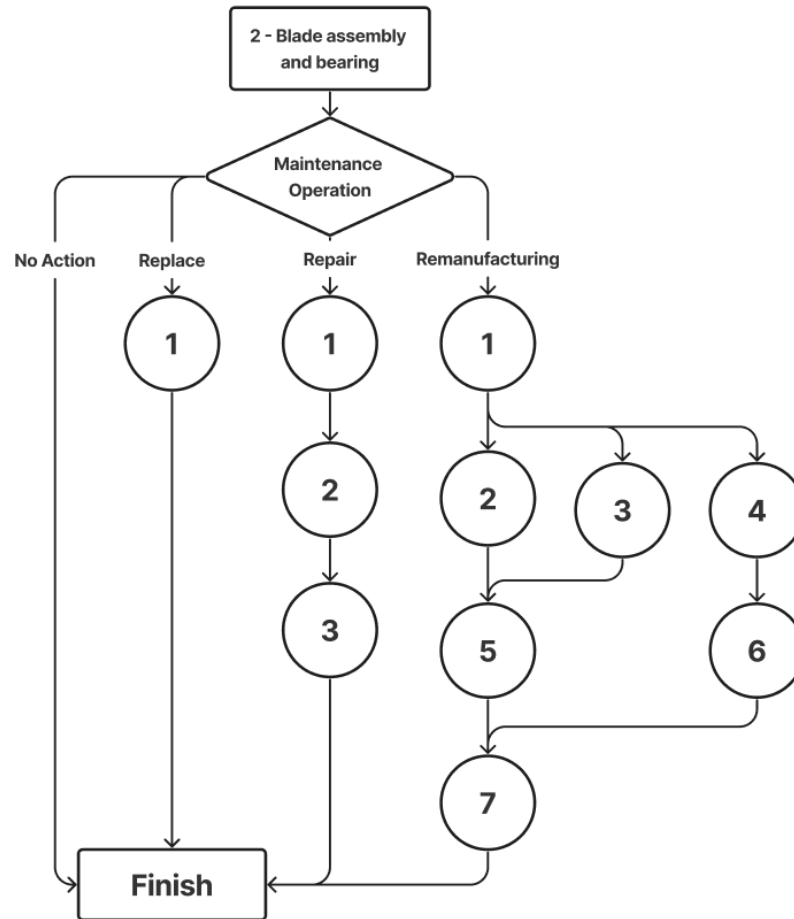


Figure 2.2: Relationship diagram for tasks and activities

In the context of our model development, a “task” is defined as a significant maintenance operation required for a component, and “activities” are the specific procedures necessary to complete that task. Understanding the difference between tasks and activities is essential for effectively planning the allocation of resources such as labor and machinery. Each task may comprise multiple activities and each activity has its own resource needs and scheduling constraints. In our study, "machines" refer to any type of equipment used in the MRO process, encompassing a wide range of tools and machinery essential for maintenance tasks. "Labor," on the other hand, includes any required technician or handler involved in the MRO activities. For the sake of simplicity in our modeling and analysis, we categorize resources into these two broad groups: machines and labor. This

approach streamlines the complexity of the model while capturing the essential aspects of resource allocation in MRO operations.

This hierarchical structure of tasks and activities directly affects the scheduling process as resource sharing between activities must be carefully managed to ensure that overlapping resource demands do not lead to scheduling conflicts.

The objective function of the mathematical model is to minimize the total cost of MRO operations. This total cost encapsulates the expenses associated with the current maintenance activities and the projected cost for future maintenance windows, the cost related to unplanned maintenance events and penalties incurred from deviations of the planned schedule. The solution of the model incentivizes early project completion with bonuses, aligning the optimization with both cost efficiency and operational excellence. Resource allocation is a pivotal component of the model. The limitations of available labor and machinery resources, and the necessity for their sharing across various tasks, are integral constraints within the model that significantly affect scheduling decisions. The model ensures that the finite availability of these resources is accounted for in the scheduling, thereby preventing over-allocation and ensuring that maintenance activities proceed without unnecessary delays.

Remanufacturing is considered a strategic option due to its potential to reduce cost and downtime compared to acquiring new parts. By integrating remanufacturing into the model, we provide a more flexible and comprehensive approach to MRO, which can adapt to a variety of operational circumstances and contribute to sustainable maintenance practices.

2.3.2 Cost Analysis of Maintenance Operations

This section elaborates on the primary cost components of the MRO scheduling model. We consider the cost incurred in the current maintenance window, anticipated cost for the next maintenance window, unplanned maintenance cost, and penalty cost or potential bonuses in the objective function of the model. All symbols and parameters used in this section are comprehensively defined as follow:

Parameters and notations used in this section:

CPN: Set of components

TSK : Set of tasks

C_{jr} : Cost of performing task r on component j in the current maintenance window

LC_{jr} : Labor cost for task r in component j

MC_{jr} : Machine cost for task r in component j

PC_{jr} : Material cost for task r in component j

LU_{jrk} : Number of labors needed for activity k in task r in component j

MU_{jrk} : Number of machines needed for activity k in task r in component j

D_{jrk} : Duration for activity k of task r in component j

HLC : Hourly labor cost

HMC : Hourly machine cost

ECN_{jr} : Estimated cost of the next maintenance of component j if task r is performed on it now

CN_{jr} : Cost for component j in the future maintenance window if task r is performed now

PN_{jr} : Probability of not failing before the next maintenance for component j if task r is done

ECU_j : Estimated cost of unplanned maintenance of the component j

CU_j : Cost of unplanned maintenance for component j

PU_{jr} : Probability of failing before the next maintenance for component j if task r is performed

FD_j : Flight delay cost caused by an unplanned maintenance of component j

DOS_j : Time that the airplane is out of service due to unplanned maintenance of component

HFD : Flight delay cost for each hour caused by unplanned maintenance

TT : Transport time for moving airplane to the maintenance facility

ST : Slack time

NS : Number of seats of the airplane

SP : Price of a seat for a day

FP : The percentage of the seats that is expected to be full for the flight

The immediate cost for the current maintenance window, denoted as C_{jr} , is predicated on the maintenance tasks selected for each component. This encompasses the expenses for materials, labor and machinery required to execute these tasks. These cost items are calculated using the Equations (1) to (3) below. Equation (1) articulates C_{jr} as the sum of the labor cost LC_{jr} , machine cost MC_{jr} , and material cost PC_{jr} for task r on component j . The labor cost in Equation (2) and

machine cost in Equation (3) are calculated by multiplying the hourly cost of labor (HLC) and machinery (HMC), respectively, by the duration (D_{jrk}) and the quantity of resources needed (LU_{jrk} for labor units and MU_{jrk} for machine units) for each activity of a task. These hourly rates are ascertained based on industry standards or contractual agreements.

$$C_{jr} = LC_{jr} + MC_{jr} + PC_{jr} \quad \forall j \in CPN, r \in TSK \quad (1)$$

$$LC_{jr} = HLC \times \sum_{k \in ACT_{jr}} (D_{jrk} \times LU_{jrk}) \quad \forall j \in CPN, r \in TSK \quad (2)$$

$$MC_{jr} = HMC \times \sum_{k \in ACT_{jr}} (D_{jrk} \times MU_{jrk}) \quad \forall j \in CPN, r \in TSK \quad (3)$$

Replacing a component can be costly due to high price of new parts, resulting in a substantial component cost, PC_{jr} . Remanufacturing may involve larger number of activities and longer completion time compared to straightforward replacement; it is considered as a cost-effective alternative in developing the model. This is attributed to the fact that the combined operation and material costs of remanufacturing are generally lower than acquiring new components, making it economically beneficial even with the increased amount of work and labor force.

For the anticipated costs in the next maintenance window, ECN_{jr} , as calculated in Equation (4) below, the model takes into account the quality of the current maintenance and its effect on the longevity of the component. This is quantified by the probability of planned maintenance, PN_{jr} , the likelihood that the component will function reliably until the next scheduled maintenance. High-quality maintenance work performed in the current period can lead to better component conditions and reduce the risk of failures to result in lower anticipated cost for future maintenance. These probabilities are based on maintenance task outcomes, historical performance data, and predictive analytics.

$$ECN_{jr} = CN_{jr} \times PN_{jr} \quad \forall j \in CPN, r \in TSK \quad (4)$$

The estimated cost of unplanned maintenance, denoted by ECU_j in Equation (5) below, is the potential financial burden when a component fails before the next scheduled maintenance. This estimate is not the actual cost but rather a projection, factoring in the likelihood of such an event. The cost of the unplanned maintenance, CU_j , in Equation (6) below, is the sum of the flight delay costs (FD_j), required labor (LC_{jr}), machinery (MC_{jr}) and material costs, and component

replacement cost (PC_{jr}). Replacement is often the only response option in urgent situations due to its immediacy in restoring operational capacity.

$$ECU_j = CU_j \times PU_{jr} \quad \forall j \in CPN, r \in TSK \quad (5)$$

$$CU_j = FD_j + LC_{jr} + MC_{jr} + PC_{jr} \quad \forall j \in CPN, r \text{ is Replace} \quad (6)$$

The calculation of the unplanned maintenance cost follows that proposed in [\(Y. Liu et al., 2019\)](#). It accounts for the additional overheads incurred by unexpected maintenance tasks and the pressing need to address them.

Flight delay cost (FD_j), a significant portion of CU_j , reflects the operational disruptions caused by unplanned maintenance. It includes revenue loss and logistical readjustments and will be further illustrated later in the numerical example discussed in Section 2.4 of this thesis.

Both the estimated unplanned maintenance cost (ECU_j) and the estimated cost for the next maintenance window (ECN_{jr}) depend on component failure probability driven from the performed maintenance tasks. The quality of these tasks directly affects the component failure probability and hence the estimated cost. These probability values are estimated based on historical data and specific maintenance actions and will be illustrated through the numerical example in section 2.4.

Equation (7) below shows the probability of unplanned maintenance (PU_{jr}) as the inverse of the probability of planned maintenance (PN_{jr}).

$$PU_{jr} = 1 - PN_{jr} \quad \forall j \in CPN, r \in TSK \quad (7)$$

2.3.3 Project Cost Optimization Model

In developing the mathematical model, the primary objective is to minimize the overall cost of the MRO project. The MRO project completion time is critical in determining this cost. Delays beyond the due date incur a penalty cost for each unit of time that exceeds the deadline, while early completion earns a bonus for each unit of time saved. The integer programming model presented below includes various constraints to ensure the proper sequencing of tasks and activities while respecting resource availability. These constraints encompass considerations for two types of limited resources: labor and machines shared among all activities and tasks. Material resources are

assumed unlimited and available as needed. The model also accommodates task interdependencies. Tasks may be interrupted, whereas activities cannot be stopped once initiated. Other key assumptions in developing the model include treating remanufactured components as equivalent to new components, assuming that replaced components will function fully as designed. Repair is assumed to enhance a component quality to the next status. For essential components, post-maintenance failure probabilities must be below certain established threshold.

The sets, parameters and variables used in the proposed mathematical model are listed below.

Sets

TME : Set of time periods

CPN : Set of components

RES : Set of resource types

TSK : Set of tasks

ACT_{jr} : Set of activities of task r in component j ($r \in TSK, j \in CPN$)

Parameters

T : Final time in the planning horizon

R : Number of the available maintenance options

J : Number of the components in the work package

K_{jr} : Number of the activities in task r for component j

C_{jr} : Cost of performing task r on component j in the current maintenance window

PN_{jr} : Probability of not failing before the next maintenance for component j if task r is performed

D_{jrk} : The duration for activity k of task r in component j

ECU_j : Estimated cost of unplanned maintenance for component j

ECN_{jr} : Estimated cost for component j in the future maintenance if task r is performed

$ES_j = \begin{cases} 1, & \text{if component } j \text{ is essential} \\ 0, & \text{otherwise} \end{cases}$

DD : Due date of the project

DC : Cost of the project delay per each unit of time

B : Bonus for finishing the project sooner per unit of time

AR_l : Available resource of type l

RN_{jrkl} : Amount of resource l needed for activity k of task r in component j

MR : Minimum acceptable reliability for essential components

M : A large value (Big M)

Decision variables

$$S_{jrt} = \begin{cases} 1, & \text{if component } j \text{ maintenance starts at time } t \text{ with task } r \\ 0, & \text{otherwise} \end{cases}$$

$$A_{jrkt} = \begin{cases} 1, & \text{if activity } k \text{ of task } r \text{ in component } j \text{ starts at time } t \\ 0, & \text{otherwise} \end{cases}$$

The integer programming model for the considered MRO operation scheduling is presented below with the parameters and variables defined above.

$$\begin{aligned} \text{Min } & \sum_{j=1}^J \sum_{r=1}^R \sum_{t=1}^T (C_{jr} + ECU_j + ECN_{jr}) \times S_{jrt} + DC \times \sum_{r=1}^R \sum_{t=DD}^T (t - \\ & DD) \times S_{jrt} - B \times \sum_{r=1}^R \sum_{t=1}^{DD} (DD - t) \times S_{jrt} \end{aligned} \quad (8)$$

Subject to:

$$\sum_{r=1}^R \sum_{t=1}^T PN_{jr} \times S_{jrt} \geq MR \quad j \in CPN \text{ and } ES_j = 1 \quad (9)$$

$$\sum_{r=1}^R \sum_{t=1}^T S_{jrt} = 1 \quad j \in CPN \quad (10)$$

$$\sum_{t=1}^T A_{jrkt} \leq 1 \quad j \in CPN, r \in TSK, k \in ACT_{jr} \quad (11)$$

$$\sum_{t=1}^T S_{jrt} \times t \leq \sum_{t=1}^T A_{jrkt} \times t \quad j \in CPN, r \in TSK, k \in ACT_{jr} \quad (12)$$

$$\sum_{k=1}^{K_{jr}} \sum_{t=1}^T A_{jrkt} \leq \sum_{t=1}^T S_{jrt} \times M \quad j \in CPN, r \in TSK \quad (13)$$

$$\sum_{r=1}^R \sum_{t=1}^T S_{irt} \times t \leq \sum_{r=1}^R \sum_{t=1}^T S_{jrt} \times t \quad i, j \in CPN \text{ with } < i, j > \quad (14)$$

$$\sum_{t=1}^T (t + D_{irk}) \times A_{irkt} \leq \sum_{q=1}^R \sum_{t=1}^T S_{jqk} \times t \quad i, j \in CPN \text{ with } < i, j >, r \in TSK, k \in ACT_{ir} \quad (15)$$

$$\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} \leq \sum_{t=1}^T A_{jrkt} \times t \quad j \in CPN, r \in TSK, a, b \in ACT_{ir} \text{ with } < a, b > \quad (16)$$

$$\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq \max(1, t - D_{jrkt})} RN_{jrkl} \times A_{jrkw} \leq AR_l \quad t \in TME, l \in RES \quad (17)$$

The objective function in Eq.(8) is to minimize the total project cost including current maintenance cost, future maintenance cost, penalty cost for project delay, and bonuses for early completion. Eq.(9) ensures that the reliability of essential components does not fall below the minimum acceptable level. Eq.(10) guarantees that all components undergo one of the available options, including "No Action." Eq.(11) enforces that each activity is performed only once. Eq.(12) ensures that all activities within a task commence after the task starting time. Eq.(13) states that if a task is not selected, none of the activities within that task is performed. Constraints in Eqs.(14), (15), and (16) capture the interconnections among components and activities. Eq.(17) enforces resource limitations.

The developed model allows tasks to be interrupted, as it does not impose constraints on their finish times but rather indicates the start time of the task. The task will be considered finished if all of its activities are completed. This characteristic is reflected in Eq.(15) to guarantees that a task for a component cannot start unless all the activities of a precedent component are already finished.

Solving the model to optimality is NP-hard due to the combinatorial complexity introduced by binary integer variables. The amount of search generally leads to an exponential increase in solution space with the size of the problem. While this classification implies significant computational challenges for large-scale problems, we contend that the model remains computationally manageable for small-scale instances. This assertion is based on preliminary tests with off-the-shelf optimization software demonstrating that optimal or near-optimal solutions can be found within a reasonable timeframe for such problem sizes. Detailed results and analysis of these tests are presented in the next section with computational performance of the developed mathematical model.

2.4 An MRO Scheduling Example with Different Scenarios

This section presents a numerical example to illustrate the effectiveness of the proposed MRO scheduling model. The example is grounded in data from a predictive maintenance study by (Verhagen & De Boer, 2018), focusing on aircraft components with higher-than-average unscheduled removal rates (URR). The data in this study provide insights into operational factors influencing component reliability which is an essential aspect of predictive and descriptive

capabilities of the model proposed in this research. In (Verhagen & De Boer, 2018), time-dependent and time-independent Proportional Hazard Models (PHMs) were utilized to analyze historical operational and maintenance data and to identify operational factors that measurably affect the occurrence of maintenance events. Their findings suggest that revised maintenance schedules based on these reliability models can notably decrease unscheduled maintenance occurrences.

For simplicity, we consider 7 components in this example problem although actual maintenance projects typically involve larger number of components. These components are selected for their deviation from the global fleet average URR with a heightened likelihood of unscheduled maintenance events. The status of each component, assigned a random tag from 1 to 5 (with 1 being the best condition and 5 the worst), is only determined post-inspection to reflect real-world uncertainty in component conditions. The probability of failure for each component correlates with its status tag.

We assume that replacing or remanufacturing a component is to restore it to its original state. Repair, however, is considered to only improve the component condition by one level. Essential components, identified using the concept of censoring from statistics, have a censoring rate above 90%, indicating their critical nature and the preference for replacement even if not fully dysfunctional. In aviation, censoring is a common practice for safety-critical components, ensuring preventative maintenance before actual failures.

Cost calculations for each maintenance task follow Equation (1) from Section 2.3, incorporating the labor, machinery, and material costs associated with each task and component condition.

2.4.1 Data Sets and Scenarios

In this section, we discuss in more details on certain key parameters from the data used in this study and explain their calculations. The diagram of the network for this project given in Figure 2.2 in Section 2.3, shows the interdependencies among the components. For clarity, we include a start node and completion node in the network as dummy components. These 2 nodes are assigned with zero duration, cost, and resource requirements. The number of activities for each component

varies depending on the chosen maintenance task. Figure 2.2 in Section 2.3 exemplifies this relationship for Component 2.

To demonstrate the model, we use a consistent dataset in different scenarios and vary certain parameters. Determining exact benchmarks for hourly labor cost (*HLC*) and machine cost (*HMC*) poses challenges due to factors of geographic locations, maintenance work types and machinery complexity. In this study, we use an industry average of \$30/hour for both *HLC* and *HMC*, aligning with the average hourly wage for mechanics in support activities in air transportation sector as shown in (U.S. Bureau of Labor Statistics, n.d.). These values are general estimates and may vary in different situations.

The material cost (PC_{jr}) can vary significantly depending on the specific component in question. For simplicity, in this example, we assign 10% and 30% of new component price, respectively, for repair and for remanufacturing. This is based on realistic estimates of material cost in MRO processes. For replacement tasks, the material cost is assumed to be the full price of a new component.

Table 2.1 Values for component 2 in Scenario 1 below provides detailed information pertaining to Component 2 in Scenario 1. It enumerates the various activities associated with each task, along with their respective durations and resource requirements. Additionally, this table calculates the labor cost (*LC*), machine cost (*MC*), and material cost (*PC*) for each activity, taking into account the specific durations and resources needed for these tasks.

Table 2.1 Values for component 2 in Scenario 1

Sample Component	Task	Activity (Type)	Duration (Hours)	Labor (Number)	Machine (Number)	Labor Cost	Machine Cost	Total Resource Cost
2-697,071,003: Blade assembly and bearing	No Action	0	0	0	0	\$0	\$0	\$0
	Replace	1	12	1	1	\$360	\$360	\$720
	Repair	1	2	1	1	\$60	\$60	\$2,220
		2	14	2	0	\$840	\$0	
		3	21	1	1	\$630	\$630	
	Remanufacturing	1	2	1	1	\$60	\$60	\$20,430
		2	75	3	2	\$6,750	\$4,500	
		3	32	2	1	\$1,920	\$960	
		4	14	1	0	\$420	\$0	
		5	22	2	2	\$1,320	\$1,320	
		6	10	1	0	\$300	\$0	
		7	47	1	1	\$1,410	\$1,410	

In our study, addressing unplanned maintenance that arises from aircraft operations is a key factor. When maintenance is required outside the scheduled window, it often leads to flight cancellations, bringing significant financial implications for airlines. Earlier, we introduced the calculation of the costs associated with such unplanned maintenance using Equation (6). Now, we aim to delve deeper into the flight delay cost (FD_j), explaining it in more detail using the formulas set out in Equations (18) to (20).

$$FD_j = DOS_j \times HFD \quad \forall j \in CPN \quad (18)$$

$$DOS_j = TT - ST + D_{jr1} \quad \forall j \in CPN, r \text{ is Replace} \quad (19)$$

$$HFD = \frac{NS \times SP \times FP}{24} \quad (20)$$

In Equation (18), HFD represents the Hourly Flight Delay cost, while DOS_j indicates the total time that a flight is delayed due to unplanned maintenance for component j . This total delay time is comprehensively calculated by considering several factors: the time required to transport (TT) the aircraft to a maintenance facility (if needed), the duration of the maintenance operation itself (D_{jr1}), and a pre-allocated slack time (ST). Slack time serves as a buffer within the maintenance schedule to accommodate any delay up to a certain threshold.

Specifically, D_{jr1} in Equation (19) refers to the duration of the replacement activity for component j during unplanned maintenance. In our example, we assume that each unplanned maintenance instance for a component consists of only one activity – replacement – due to the urgency of these situations.

In our study, understanding the financial implications of unplanned maintenance on airline operations is pivotal, and this is particularly evident when considering the HFD . According to (Cook et al., 2004), the average cost of flight delays is estimated at \$78 per minute, which translates to an HFD of approximately \$4680. This cost encompasses a range of factors, including crew salaries, handling agent fees, airport charges, and the costs related to passenger delays and network disruptions.

To delve deeper into the varying impacts of unplanned maintenance, we adopt a systematic approach to calculate *HFD*. (Cook et al., 2004) provide a specific estimate for flight delay cost, but our study aims to explore a range of scenarios, in one which we are adjusting the *HFD* values. This is facilitated through Equation (20), where *HFD* is calculated based on the expected daily revenue per seat, multiplied by the number of seats in the aircraft, and adjusted for the expected occupancy rate. Such a method allows for a flexible examination of different scenarios by varying parameters like the seat price or the seat occupancy rate.

In the initial scenario of our study, we set the parameters in a way that our calculated *HFD* closely aligns with (Cook et al., 2004) estimate. For an Airbus A330 with 345 seats, our calculated cost of flight delay due to unplanned maintenance is approximately \$77.27 per minute. This calculation, based on the parameters which we will present in Table 2.2 Parameters used in in HFD calculation in scenario 1, ensures that our base scenario accurately reflects the real-world costs associated with flight delays.

Table 2.2 Parameters used in in HFD calculation in scenario 1

Transport time (TT)	10 hours
Slack time (ST)	3 hours
Aircraft Type	Airbus A330
Number of Seats (NS)	345
Price of a Seat for a day (SP)	\$430
Percentage full (FP)	75%
Flight delay cost/day of unplanned maintenance	\$111,262
Flight delay cost/hour of unplanned maintenance (HFD)	\$4,635
Flight delay cost/minute of unplanned maintenance	\$77

Table 2.3 Component information and unplanned maintenance cost in our study provides comprehensive details about the components considered in the analysis. This includes each component's price, condition, censoring percentage, its designation as an essential component, and the associated cost of unplanned maintenance. A notable aspect in Table 2.3 Component information and unplanned maintenance cost is Component 6, which can be replaced directly at the site of operation. This on-site replacement capability eliminates the necessity for transporting the component to a maintenance facility, effectively reducing the transport time (*TT*) for this component to zero.

Table 2.3 Component information and unplanned maintenance cost

Components	Unit Price	Condition	Censoring (%)	Essential	Cost
1-Dummy Start	\$0	1	100	No	\$0
2-697,071,003:Blade assembly and bearing	\$57,898	5	75	Yes	\$161,039
3-1,152,106-3:DC starter generator	\$51,076	3	78	Yes	\$137,866
4-3-1573-1: MLG wheel and tire assembly	\$64,529	3	93	Yes	\$157,669
5-3-1574: NLG wheel and tire assembly	\$25,468	2	94	Yes	\$106,807
6-728,809-1: Thermal actuator	\$125	5	36	No	\$22,107
7-Dummy Finish	\$0	1	100	No	\$0

Table 2.4 in our document presents crucial information used to estimate the costs for the next maintenance window. This table outlines potential statuses of components following the current maintenance cycle and the probable maintenance operations that might be needed in the future. For instance, if a component's status post-maintenance is marked as 1, there's a 90% likelihood that no further action will be required in the next maintenance window, with a 10% chance of needing a repair. To estimate the future maintenance costs, we calculate a weighted average cost, considering the probabilities and associated costs detailed in Table 2.4 Expected maintenance tasks in the next maintenance window.

An important assumption in our model, as indicated in Table 2.4 Expected maintenance tasks in the next maintenance window, is that components left in status 5 after maintenance are deemed beyond repair or remanufacturing. In such cases, replacing the component becomes the sole feasible option. Table 2.4 Expected maintenance tasks in the next maintenance window also includes a column that lists the failure probabilities for each component, determined based on their respective statuses post-maintenance.

Table 2.4 Expected maintenance tasks in the next maintenance window

Status	No Action	Replace	Repair	Remanufacturing	Failure Probability
1	90%	0%	10%	0%	1%
2	10%	30%	30%	30%	10%
3	0%	40%	20%	40%	25%
4	0%	70%	0%	30%	58%
5	0%	100%	0%	0%	95%

Table 2.5 Current and future maintenance costs details the costs associated with the maintenance of each component. This includes both the current costs incurred in the ongoing maintenance window and the anticipated costs for future scheduled maintenance tasks. Notably, the table reveals that certain components, specifically Components 5 and 6, lack the option for remanufacturing. This exclusion is due to either the relative low cost of these components, making

remanufacturing economically unfeasible, or existing technological and design constraints that impede the remanufacturing process. Furthermore, the table indicates that the repair option is unavailable for Component 5. This is attributed to the higher costs and reduced quality of repair in comparison to replacing the component entirely.

Table 2.5 Current and future maintenance costs

Component	No action		Replace		Repair		Remanufacturing	
	Current	Future	Current	Future	Current	Future	Current	Future
1	0	0	0	0	0	0	0	0
2	0	\$58,618	\$58,618	\$8009	\$8,009	\$52,372	\$37,799	\$8009
3	0	\$33,245	\$51,616	\$8887	\$8,887	\$26,266	\$27,052	\$8887
4	0	\$55,349	\$66,029	\$1,263	\$12,632	\$35,397	\$30,098	\$1,2632
5	0	\$12,777	\$25,948	\$5996	\$5,996	\$5996	NA	\$5996
6	0	\$545	\$545	\$545	NA	\$545	NA	\$545
7	0	0	0	0	0	0	0	0

Table 2.6 Failure probability for each component depending on the task performed on it presents the failure probabilities for each component, key indicators for assessing maintenance effectiveness. It's important to note that while the values in this table are arbitrarily selected, they are chosen to logically reflect the outcomes of various maintenance tasks performed on the components. This logical flow ensures that the probabilities align realistically with the maintenance actions, providing a meaningful basis for evaluating potential risks and guiding future maintenance decisions.

Table 2.6 Failure probability for each component depending on the task performed on it

Component	No Action	Replace	Repair	Remanufacturing
1	0%	0%	0%	0%
2	95%	1%	58%	1%
3	25%	1%	10%	1%
4	25%	1%	10%	1%
5	10%	1%	1%	1%
6	95%	1%	58%	1%
7	0%	0%	0%	0%

To evaluate the model's performance, we consider five different scenarios. The first scenario is set as the base case. This serves as a benchmark for comparison and analysis across the scenarios. Notably, in these scenarios, there are no restrictions on materials for the analyses. However, resources such as labor and machinery are limited. These resource constraints are consistent across each scenario.

2.4.2 Operational Performance Analysis

In our study, we rigorously analyze the model's performance through five distinct scenarios, each carefully chosen to mirror real-world considerations in MRO operations. The scenarios are designed to evaluate how varying key parameters—such as resource availability, unplanned maintenance costs, delay penalties, and remanufacturing options—impact maintenance strategies and decisions. This approach enables us to gain a comprehensive understanding of the model's applicability and flexibility in different operational contexts. Below, we delve into the real-world relevance of each scenario parameter:

1. **Resource Availability:** This aspect reflects the common constraints faced in MRO facilities, where labor and machinery are often limited. By varying the number of available resources in different scenarios, we can assess how these limitations impact the scheduling and efficiency of maintenance tasks.
2. **Cost of Unplanned Maintenance:** Unplanned maintenance is a frequent and costly challenge in aviation, leading to operational disruptions and financial burdens. By altering the cost of unplanned maintenance in our scenarios, we investigate its influence on the selection of maintenance strategies and the overall financial impact on operations. This aspect is particularly relevant in assessing the model's capacity to adapt maintenance plans in response to fluctuating maintenance costs.
3. **Penalties for Delays:** In the real world, maintenance delays can lead to substantial penalties for airlines due to disrupted schedules. Including varying penalty rates in our scenarios helps us understand the trade-offs between maintenance costs and the urgency of completing tasks on time.
4. **Availability of Remanufacturing Options:** Remanufacturing is increasingly recognized for its role in sustainable MRO practices. By including scenarios where remanufacturing is either available or not, we evaluate its impact on maintenance decision-making, cost-effectiveness, and resource management. This approach enables us to assess the potential of remanufacturing in enhancing sustainability and operational efficiency in MRO activities.

The scenarios are designed to reflect a range of real-world conditions, from constrained resources to varying costs and operational challenges. By employing the CPLEX optimization software, a

robust tool widely used in industry and academia, we ensure the reliability and relevance of our solutions. This software's capacity to handle complex optimization problems allows us to effectively navigate the multifaceted nature of MRO scheduling under diverse operational circumstances. This analytical approach offers valuable insights into optimal maintenance strategies, aiding decision-makers in navigating the complexities of real-world MRO operations.

Scenario 1: Detailed Analysis and Interpretation

Scenario 1 sets the baseline for our study, featuring a balanced allocation of four units each for labor and machinery resources. The penalty and bonus rates are uniformly set at \$500 per hour for any delays or early completion, respectively. This scenario's results, pivotal in understanding the model's decision-making under constrained resources and cost considerations, are elaborately presented in Table 2.7 Results for scenario 1 (and scenario 4).

Table 2.7 Results for scenario 1 (and scenario 4)

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	35	47
3 -1,152,106–3: DC starter generator	Repair	Activity 1	1	4
		Activity 2	48	68
		Activity 3	70	90
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	5	39
		Activity 3	40	90
5- 3–1574: NLG wheel & tire assembly	No Action	-	90	90
6- 728,809–1: Thermal actuator	Replace	Activity 1	6	13
7- Dummy Finish	No Action	-	90	90

Key Decisions and Rationale:

Component 2 - Blade Assembly and Bearing (Replacement): This component is selected for replacement due to its critical condition, denoted by status 5, which indicates a 95% chance of failure before the next maintenance cycle. The decision to replace rather than remanufacture is informed by a strategic balance between cost and project duration. While replacement incurs higher immediate costs, it significantly reduces the maintenance timeline, offsetting potential penalties associated with extended project duration. This choice is not influenced by the quality or unplanned maintenance costs, as both replacement and remanufacturing would restore the

component to full functionality. However, the urgent need to minimize project delays and associated costs makes replacement the preferred option in this scenario, highlighting the model's capacity to make decisions that prioritize critical maintenance tasks within the constraints of time and resources.

Component 3 - DC Starter Generator (Repair): The decision is guided by the trade-offs between cost, quality, and project timeline. This component, assigned a status less critical than that of Components 2 and 6, offers more flexibility in maintenance options. The model opts for repair over replacement or remanufacturing. This choice is driven by two primary factors: the projected improvement in component quality and the impact on the maintenance schedule.

Repairing Component 3 efficiently upgrades its status, significantly reducing the likelihood of future failures while being more cost-effective than replacement. Moreover, remanufacturing, although a viable option in terms of quality improvement, is not selected due to the longer duration required for this process, which would result in higher penalty costs due to project delays. The model's decision to repair Component 3 demonstrates a balanced approach, considering both the need to improve component quality to a satisfactory level and to maintain the efficiency of the maintenance schedule, thus optimizing overall project costs.

Component 4 - MLG Wheel & Tire Assembly (Remanufacturing): It illustrates the strategic selection process in the model. This component, with a status of 3, presents a moderate risk of failure. The model chooses remanufacturing over replacement or repair, a decision influenced by a combination of factors related to cost, risk management, and project timing.

Remanufacturing Component 4 effectively reduces its failure risk to an acceptable level, making it a particularly suitable option given that this component is essential and safety-sensitive. The process brings the component's quality close to that of a new part but at a lower cost than outright replacement. Although remanufacturing requires more time than replacement, in this scenario, the trade-off between the extended duration and the reduced material cost is deemed beneficial.

The decision to opt for remanufacturing also reflects a consideration of the available resources and the overall project timeline. By choosing a process that is longer but less expensive, the model demonstrates a preference for a cost-effective approach that still meets the stringent safety

requirements for an essential component. The remanufacturing of Component 4 showcases the model's ability to balance multiple factors — cost efficiency, risk mitigation, and adherence to safety standards — in the maintenance scheduling process.

Component 5 - NLG Wheel & Tire Assembly (No Action): Here, the decision-making process in the model illustrates a different aspect of maintenance scheduling, where it opts for no action. This decision is based on the component's relatively favorable status (rated as status 2), indicating a lower risk of failure before the next scheduled maintenance window.

In this case, the model assesses that the condition of Component 5 does not warrant immediate intervention, thereby saving on unnecessary maintenance costs and resources. By opting for no action, the model effectively allocates resources to other components that require more urgent attention or where maintenance actions would significantly reduce the risk of failure.

This decision underscores the model's capability to not only identify components needing maintenance but also to recognize when maintenance can be safely deferred. It reflects a strategic approach to maintenance scheduling, focusing resources and efforts where they are most needed, based on the condition of each component and the overall priorities of the maintenance project. By avoiding unnecessary maintenance on Component 5, the model ensures more efficient use of resources and minimizes potential disruptions to the maintenance schedule.

Component 6 - Thermal Actuator (Replacement): the decision to replace Component 6 is driven by two key factors: the component's condition and the available maintenance options. Component 6 is in a critically poor condition (status 5), indicating a very high risk (95%) of failure before the next scheduled maintenance. This level of risk necessitates immediate action to prevent potential failure and ensure the safety and reliability of the aircraft.

Moreover, for Component 6, the only available maintenance option is replacement. Unlike other components where multiple maintenance actions such as repair or remanufacturing might be applicable, Component 6's design or technical constraints limit the available maintenance strategies exclusively to replacement. This restriction in available maintenance options means that when faced with a component in such poor condition, the model defaults to replacement as the only feasible action.

The choice to replace Component 6, despite its higher cost compared to other maintenance actions, is therefore a reflection of both its poor condition and the lack of alternative maintenance strategies. This decision underscores the priority of maintaining operational safety and reliability, particularly for components in critical condition where the risk of failure is unacceptably high.

Scenario 2: Impact of Resource Availability on Maintenance Decisions

In Scenario 2, the study delves into how variations in resource availability influence maintenance task selection, project delays, and overall costs. This scenario is split into two parts: Scenario 2A with reduced resources (2 units each of labor and machinery) and Scenario 2B with increased resources (8 units each).

Scenario 2A - Reduced Resources (2 Units Each): This scenario, as depicted in Table 2.8 Result for scenario 2A (2 unit of resources), examines the model's response to constrained resource availability. The key findings include:

Table 2.8 Result for scenario 2A (2 unit of resources)

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	74	86
3 -1,152,106–3: DC starter generator	Replace	Activity 1	81	910
4- 3–1573–1: MLG wheel & tire assembly	Repair	Activity 1	1	5
		Activity 2	6	72
5- 3–1574: NLG wheel & tire assembly	No Action	-	72	72
6- 728,809–1: Thermal actuator	Replace	Activity 1	73	80
7- Dummy Finish	No Action	-	90	90

Component 2 (Replacement): The model continues to prioritize replacement over remanufacturing, a decision influenced by its critical condition (status 5) and the high risk of failure. With even fewer resources, remanufacturing becomes less favorable due to its longer duration and higher resource demand.

Component 3 (Replacement): Unlike Scenario 1, where repair was chosen, this component is now selected for replacement. The shift from repair to replacement under limited resources is a strategic decision to mitigate the increased risk of project delays and associated costs. This highlights how resource scarcity can lead to prioritizing quicker but potentially costlier maintenance actions.

Component 4 (Repair): In a departure from Scenario 1's remanufacturing choice, repair is selected due to its lower resource intensity. This decision reflects the model's adaptability in optimizing task selection to align with the available resources, ensuring essential maintenance within the constraints.

Component 5 (No Action): The decision for no action is consistent with Scenario 1, as even with reduced resources, this component's condition (status 2) does not necessitate immediate intervention, thereby conserving resources for more critical tasks.

Component 6 (Replacement): Replacement remains the sole viable option due to the component's poor condition and limited maintenance alternatives. This decision is crucial irrespective of resource availability, underscoring the need to address critical maintenance needs.

Scenario 2B - Increased Resources (8 Units Each): In contrast, Scenario 2B, shown in Table 2.9 Result for scenario 2B (8 unit of resources), explores the model's behavior with an abundance of resources. The results demonstrate a tendency to select longer, more resource-intensive tasks, potentially leading to cost savings:

Table 2.9 Result for scenario 2B (8 unit of resources)

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Remanufacturing	Activity 1	1	3
		Activity 2	3	78
		Activity 3	44	76
		Activity 4	61	75
		Activity 5	78	100
		Activity 6	86	96
		Activity 7	100	147
3 -1,152,106–3: DC starter generator	Remanufacturing	Activity 1	1	4
		Activity 2	39	47
		Activity 3	14	34
		Activity 4	97	117
		Activity 5	85	113
		Activity 6	117	147
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	5	39
		Activity 3	42	92
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	106	111
		Activity 2	112	147
6- 728,809–1: Thermal actuator	Replace	Activity 1	129	136
7- Dummy Finish	No Action	-	147	147

Component 2 (Remanufacturing): The model opts for remanufacturing, a departure from replacement in Scenario 1 and 2A. This shift can be attributed to the availability of ample resources, allowing for a more time-intensive but cost-effective maintenance action.

Component 3 (Remanufacturing): The choice of remanufacturing over repair or replacement is driven by the aim to enhance the component's quality more substantially, reducing future maintenance and unplanned maintenance costs. The abundance of resources allows for this extensive process without substantial delay penalties.

Component 4 (Remanufacturing): As in Scenario 1, remanufacturing is the chosen maintenance action, aligning with the model's objective to ensure component reliability and safety.

Component 5 (Repair): The decision to repair in Scenario 2B contrasts with the 'no action' in Scenario 2A, demonstrating the model's ability to allocate additional resources to improve the component's condition, thereby reducing future maintenance and unplanned maintenance costs without incurring high penalties.

Component 6 (Replacement): As in previous scenarios, replacement is the only option, dictated by the component's critical condition.

The comparison between Scenarios 2A and 2B reveals how resource levels play a crucial role in maintenance decision-making. Under resource constraints (Scenario 2A), the model tends to select quicker, less resource-intensive tasks, even if they might be costlier. Conversely, with ample resources (Scenario 2B), the model can afford to undertake more extensive maintenance actions like remanufacturing, which, while time-consuming, are often more cost-effective in the long run.

These observations highlight the critical influence of resource availability in MRO scheduling. The model demonstrates a capacity to adapt its maintenance strategy based on resource constraints, balancing the need to minimize project delays, manage costs effectively, and ensure the safety and reliability of aircraft components.

Scenario 3: The Impact of Increased Penalty Costs on Maintenance Decisions

In Scenario 3, the study explores the impact of significantly higher penalty costs for project delays, set at \$2000 per hour, while keeping other parameters the same as in Scenario 1. The result of the model under these new constraints is presented in Table 2.10 Result for scenario 3.

Table 2.10 Result for scenario 3

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	37	49
3 -1,152,106–3: DC starter generator	Repair	Activity 1	12	15
		Activity 2	16	36
		Activity 3	37	57
4- 3–1573–1: MLG wheel & tire assembly	Replace	Activity 1	1	11
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	12	15
		Activity 2	21	56
6- 728,809–1: Thermal actuator	Replace	Activity 1	13	20
7- Dummy Finish	No Action	-	57	57

Component 4 (Replacement): The model's decision to replace this component, rather than remanufacturing as in previous scenarios, is primarily influenced by the increased penalty for project delays. The higher cost of replacement is offset by the substantial reduction in project duration, thus minimizing the expensive penalties associated with longer project timelines. This change in decision highlights the model's sensitivity to penalty costs, where reducing the overall project duration becomes a higher priority than minimizing immediate maintenance costs.

Component 5 (Repair): The model's decision to repair Component 5 marks a notable shift from the previous scenarios, where no action was taken on this component. This change is driven by the increased penalty for project delays, which now makes the repair of Component 5 a viable option. By repairing Component 5, the model utilizes the time and resources saved from choosing replacement over remanufacturing for Component 4. This strategic reallocation of resources to repair Component 5 not only improves its quality but also mitigates the risk of future maintenance and unplanned maintenance costs, all while ensuring the project remains within the scheduled timeline. The decision to repair Component 5, instead of taking no action, is a calculated move to optimize the project's overall cost-effectiveness in light of the significantly higher penalty rates, demonstrating the model's dynamic adaptation to varying cost implications.

Component 2, 3, and 6: For these components, the maintenance actions remain consistent with Scenario 1, indicating that the heightened penalty rate did not necessitate a change in strategy for these particular components. This consistency suggests that for certain components, the decision-making process remains largely unaffected by the increased penalties, possibly due to the already optimized balance between maintenance actions, project duration, and costs.

Overall, the increased penalty rates in Scenario 3 significantly influence the model's maintenance task selection. The model demonstrates a strategic shift towards maintenance actions that enable quicker project completion to avoid steep penalties, even if it means selecting more expensive maintenance options. This scenario underscores the importance of considering penalty costs in maintenance scheduling, as they can considerably impact the decision-making process, particularly in prioritizing actions that reduce project duration.

Scenario 4: Analyzing the Impact of Unplanned Maintenance Costs

Scenario 4 delves into the implications of varying the costs associated with unplanned maintenance. Specifically, this scenario examines how adjustments in the seat price (SP) and the percentage of seats filled (FP) influence maintenance decisions. In this scenario, the SP is set to \$200 per seat, and the FP is assumed to be 50%. Consequently, the cost of unplanned maintenance per hour (HFD) decreases to \$1437.5, a significant reduction from the original value used in previous scenarios.

The results of Scenario 4 are documented in Table 2.7 Results for scenario 1 (and scenario 4), which is identical to Scenario 1 in terms of the maintenance tasks selected for each component. It suggests that the reduction in the cost of unplanned maintenance does not significantly influence the model's decision-making process in this particular case. This observation highlights that while unplanned maintenance costs are a crucial factor, they do not always dictate the choice of maintenance tasks, especially when other factors such as component condition and available resources play a dominant role.

The unchanged maintenance strategy, despite the altered HFD value, reflects the model's focus on optimizing maintenance actions based on component conditions and resource availability rather than solely on the potential costs of unplanned maintenance. This scenario illustrates the

multifaceted nature of decision-making in maintenance scheduling, where a range of factors are weighed in to determine the most effective maintenance strategy.

In summary, Scenario 4 reveals that while unplanned maintenance costs are important, they are not the sole determinant of maintenance decisions in certain situations. Instead, the model demonstrates a balanced approach, considering a combination of factors to ensure optimal maintenance scheduling and resource allocation.

Scenario 5: Examining the Effects of Removing Remanufacturing as an Option

Scenario 5 presents a unique situation where remanufacturing is not an available option for maintenance. This scenario maintains the same data parameters as Scenario 1, but it excludes remanufacturing from the range of possible maintenance actions. The results of this scenario, which are summarized in Table 2.11 Result for scenario 5, provide insights into how the exclusion of remanufacturing affects the maintenance schedule and resource allocation.

Table 2.11 Result for scenario 5

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	37	49
3 -1,152,106–3: DC starter generator	Repair	Activity 1	12	15
		Activity 2	16	36
		Activity 3	37	57
4- 3–1573–1: MLG wheel & tire assembly	Replace	Activity 1	1	11
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	12	15
		Activity 2	21	56
6- 728,809–1: Thermal actuator	Replace	Activity 1	13	20
7- Dummy Finish	No Action	-	57	57

Component 4 (Replacement): With remanufacturing off the table, the model defaults to replacement for Component 4. This decision significantly accelerates the maintenance process for this component, as replacement is less time-consuming compared to remanufacturing. The quicker turnaround time reduces the overall project duration by 33 hours, thereby also lowering the potential penalty costs associated with project delays. This choice demonstrates a strategic shift towards reducing project duration to manage costs effectively, even if it means opting for a more expensive maintenance action like replacement.

Component 5 (Repair): With the resources freed up from the quicker replacement of Component 4, the model can allocate sufficient resources to repair Component 5 without causing project delays. This scenario highlights the model's ability to reallocate resources effectively, ensuring that each component receives the necessary maintenance within the project's time constraints.

Consistency in Other Components: The maintenance tasks for Components 2, 3, and 6 remain the same as in Scenario 1, indicating that the removal of remanufacturing does not significantly alter the decision-making process for these components. Their maintenance strategies are driven by their individual conditions and the overall objectives of minimizing project duration and costs.

Overall, Scenario 5 illustrates the model's adaptability in the absence of remanufacturing. It demonstrates how the model recalibrates maintenance strategies to optimize resource use and project timelines, ensuring effective maintenance scheduling despite the constraints imposed by the unavailability of a maintenance option. This scenario underscores the importance of flexibility in maintenance planning and the model's capacity to adjust its strategies in response to changing maintenance options.

2.4.3 Cost Analysis

In the intricate landscape of MRO, the cost analysis conducted in this section of our study sheds light on the financial impact of various maintenance strategies employed across different scenarios. Figure 2.3 Cost terms and comparison, as referenced, delineates the breakdown of costs, highlighting the interplay between current maintenance costs, future maintenance implications, unexpected maintenance occurrences, and the financial repercussions of project timing in terms of bonuses and penalties.

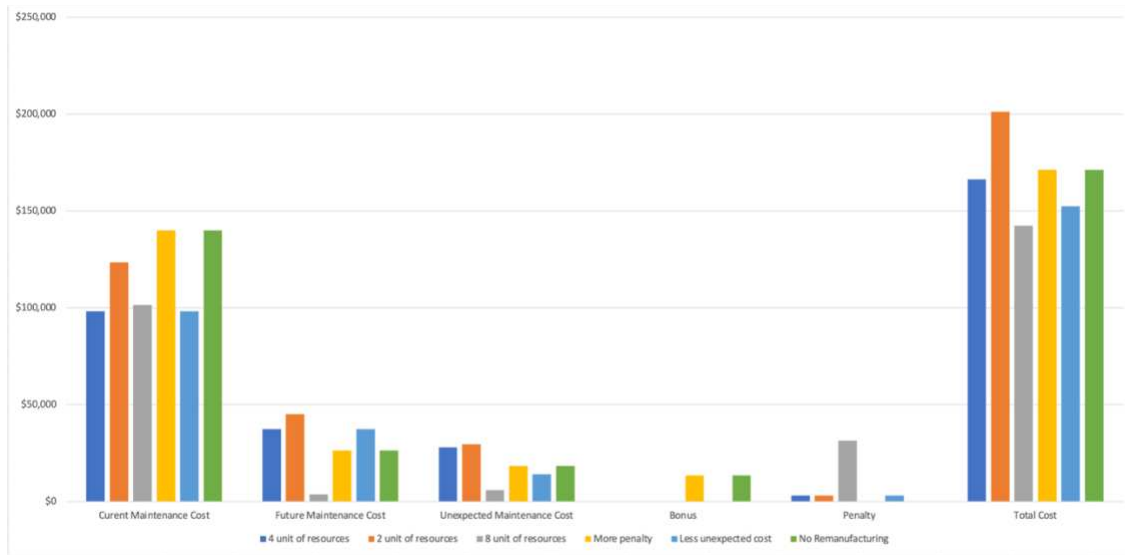


Figure 2.3 Cost terms and comparison

Scenarios 3 and 5 stand out with the highest current maintenance costs, a direct consequence of the strategic selection of maintenance actions under different penalty conditions and the availability of remanufacturing options. In Scenario 3, with penalty costs quadrupled, the model favors more expedient but expensive maintenance tasks to avoid delays. This reflects a proactive approach to mitigate penalties but at the expense of higher immediate maintenance costs. The absence of remanufacturing in Scenario 5 also elevates current maintenance costs, underscoring remanufacturing's role in cost control. Conversely, Scenarios 1 and 4 maintain lower current maintenance costs, attributable to strategic task selection that balances cost with the risk of component failure.

The decisions made today in maintenance not only affect current costs but also have long-lasting implications on component reliability. Scenario 2A, with the highest future maintenance costs, illustrates the compounding effect of limited resources leading to frequent repairs and postponed maintenance, potentially compromising component longevity. Conversely, the substantial resource availability in Scenario 2B allows for maintenance tasks that enhance component reliability, thereby reducing future costs.

The cost associated with unexpected maintenance mirrors the trends seen in future maintenance costs, with distinct variations between scenarios. In Scenario 2A, the limited resources necessitate a more cautious approach to maintenance. This constraint often leads to the selection of maintenance actions that, while less resource-intensive, may not adequately mitigate the risk of

component failure. Consequently, the model incurs higher unexpected maintenance costs due to the increased likelihood of components failing before the next maintenance cycle, reflecting the short-term risks of constrained resource allocation. In contrast, Scenario 2B benefits from a surplus of resources, which the model leverages to perform comprehensive and thorough maintenance. This wealth of resources allows for more extensive and potentially preventative maintenance actions that improve the components' condition, thereby reducing the probability of untimely failures. As a result, the robustness of the maintenance performed in Scenario 2B translates to lower unexpected maintenance costs. This reduction is indicative of the long-term benefits of resource abundance, allowing for proactive maintenance strategies that enhance overall component reliability and reduce the incidence of unexpected breakdowns.

In our cost analysis across various scenarios, the balance between early completion bonuses and delay penalties is crucial, especially highlighted in Scenarios 3 and 5. Scenario 3, with its heightened penalty rate of \$2000 per hour, drives the model to opt for quicker maintenance solutions, leading to earlier project completion and consequently earning bonuses. This scenario underscores the delicate trade-off in MRO scheduling: minimizing delay penalties can necessitate selecting faster yet more expensive maintenance actions. Conversely, Scenario 5, while maintaining the standard penalty rate, shows a reduced project duration, primarily due to the unavailability of the remanufacturing option. The absence of remanufacturing not only shortens the project timeline but also increases the current maintenance cost, as the model defaults to costlier maintenance alternatives to ensure same quality level. Scenario 2B presents an interesting contrast. Despite having the most substantial penalty costs among scenarios with penalties, it strategically utilizes its abundant resources to select more intensive tasks. This allocation effectively manages the project timeline, maintaining a balance between undertaking comprehensive maintenance actions and avoiding prohibitive delay costs. The results from Scenario 2B underscore the importance of resource availability in mitigating project delays and managing maintenance costs, particularly in scenarios where the penalty for project delays is a significant concern. This comprehensive understanding of the interplay between resource availability, maintenance actions, and penalty costs offers valuable insights into optimizing MRO scheduling for efficiency and cost-effectiveness.

When considering total costs, the study reveals that an increase in available resources correlates with a decrease in overall costs, as demonstrated by the contrast between Scenarios 2A and 2B. Moreover, Scenarios 3 and 5 incur the highest costs within their resource group due to elevated penalties and the unavailability of remanufacturing. The analysis underscores the crucial role of effective resource management and strategic planning in MRO scheduling. Decisions made in this domain have profound financial implications, not only in the immediate context but also over the longer term. These choices directly affect the ongoing reliability and future availability of aircraft components, highlighting the interconnected nature of resource allocation, cost management, and maintenance strategy.

2.4.4 Summary and Discussion

In the concluding section of our analysis, the study offers decision-making insights for the aerospace industry by examining the cost implications of different maintenance strategies under various scenarios. The results illuminate key factors that significantly influence maintenance planning and its cost-effectiveness.

Role of Remanufacturing: The analysis underscores the value of including remanufacturing as a maintenance option. Scenarios where remanufacturing was available, such as Scenario 1, consistently showed lower current maintenance costs compared to Scenario 5, where remanufacturing wasn't an option. This highlights remanufacturing's role in balancing cost-effectiveness with maintaining component quality. For the aerospace industry, this suggests a strategic focus on incorporating remanufacturing practices, which can lead to significant cost savings without compromising on component reliability.

Resource Availability: The comparison between Scenarios 2A (limited resources) and 2B (abundant resources) against the baseline Scenario 1 demonstrates that increased resource availability allows for more flexibility in choosing maintenance actions. This flexibility often translates to lower costs, as more resource-intensive but cost-effective options like remanufacturing can be selected without incurring high penalties. For maintenance planners in the aerospace industry, this emphasizes the importance of optimal resource allocation and planning to reduce costs and improve maintenance efficiency.

Impact of Penalties: The results from Scenario 3, with a quadrupled penalty rate, reveal how increased penalties for project delays influence maintenance task selection. The model tends to favor faster, albeit more expensive, maintenance actions to avoid penalties, leading to higher current maintenance costs. This highlights a critical trade-off faced in maintenance planning: balancing the urgency of project completion against the costs of maintenance actions. Decision-makers in the aerospace industry need to carefully weigh the implications of delay penalties in their maintenance planning strategies.

Future Maintenance and Unexpected Costs: The study also finds that the quality of components post-maintenance significantly affects future maintenance and unexpected maintenance costs. Scenarios 1 and 4, with similar maintenance plans, resulted in similar future maintenance requirements, although their unplanned maintenance costs differed due to different scenario-specific parameters. This indicates that maintaining a high component quality post-maintenance can lead to cost savings in future maintenance cycles, an important consideration for long-term maintenance planning in the aerospace industry.

In summary, the analysis offers valuable insights for predictive maintenance planning in the aerospace sector. It illustrates the impact of different maintenance strategies on cost-effectiveness, emphasizing the importance of remanufacturing, optimal resource allocation, and strategic decision-making in the face of penalties. These findings provide a framework for maintenance planners and decision-makers to develop more efficient, cost-effective maintenance schedules, ensuring the reliability and longevity of aircraft components.

2.5 Conclusion

In concluding this chapter, which introduces a novel integer programming model for maintenance, repair, and overhaul (MRO) processes, it's important to consider both the contributions and limitations of our approach, along with potential directions for future research.

The development of this new model represents a significant stride in optimizing MRO operations, particularly in the aerospace industry and other sectors with heavy MRO demands. Through various scenarios, the model effectively demonstrates how changes in parameters like resource availability, penalty costs, and the presence or absence of remanufacturing options can

significantly influence the selection of maintenance tasks and their associated costs. It offers a strategic tool to aid airlines and maintenance providers in making more informed decisions regarding maintenance planning and resource allocation, thereby optimizing costs and enhancing efficiency.

However, two key limitations are notable. First, the scalability of the model presents a challenge. As the size and complexity of MRO operations increase, the computational demands of the model may grow exponentially. This could limit its applicability in larger, more intricate MRO scenarios, where the need for efficient and effective planning is often most critical. Addressing this scalability concern is essential for broadening the utility of the model across a wider range of MRO projects.

Second, the accuracy and efficacy of the model are heavily reliant on the quality and precision of the input data. Inaccurate or incomplete data can significantly impair the model's ability to generate reliable and useful maintenance schedules. This dependence on data underscores the need for robust data collection and management systems in MRO operations, to ensure that the model has access to the most current and comprehensive information possible.

Looking ahead, a central direction for future research lies in enhancing the scalability of the model. While the exact formulation provides valuable insights for small-sized instances, its computational demands become prohibitive as problem size increases. To address this, the next chapter introduces a tailored Iterative Local Search (ILS) heuristic designed to extend the applicability of the proposed model to large-scale MRO scheduling problems. This heuristic framework aims to retain the core structure and decision logic of the mathematical model while providing a more computationally efficient solution approach. In doing so, it bridges the gap between theoretical modeling and practical implementation, ensuring that the benefits of optimized MRO scheduling can be realized even in highly complex, resource-constrained operational settings.

Chapter 3

An Operational-Level MRO Scheduling Optimization for Aircraft Type C and D Checks Using Iterative Local Search

Building on the modeling foundations and insights developed in the previous chapter, this chapter introduces an operational-level MRO scheduling framework that leverages Iterative Local Search (ILS) to handle large-scale, complex maintenance tasks, especially when remanufacturing is considered alongside repair and replacement strategies. The growing demand for air travel has amplified the need for more cost-effective and efficient approaches to Maintenance, Repair, and Overhaul (MRO), rendering traditional manual scheduling inadequate for high-volume operations. Moreover, as airlines seek to enhance sustainability and reduce operational costs, remanufacturing presents an attractive option for extending component life cycles—yet poses new challenges in the scheduling process. To address these challenges, this chapter develops and evaluates an optimization model that seamlessly integrates remanufacturing into MRO scheduling. Unlike purely manual or classical optimization approaches, which often become computationally prohibitive for large-scale problems, the proposed method employs an Iterative Local Search (ILS) heuristic. This heuristic systematically refines an initial solution by exploring neighboring schedules and iteratively improving upon them, thereby generating solutions that are optimal or near-optimal in a fraction of the time required by exact methods. The computational experiments presented in this chapter span both small-scale and large-scale instances, demonstrating the practical adaptability of the ILS approach. Results show that the ILS heuristic not only lowers computational time but also consistently attains high-quality solutions across various scenarios, including those that incorporate remanufacturing options. By reducing reliance on newly manufactured components, airlines can streamline resource utilization, cut costs, and bolster sustainability in their maintenance workflows. In addition to its practical focus, this chapter contributes to original research on operational-level scheduling by explicitly embedding remanufacturing decisions within a heuristic-based framework. As one of the few studies to unite these elements, it demonstrates a feasible alternative to purely mathematical optimization for real-world MRO environments. Taken together, the methods and findings herein build upon the fundamental model from the previous chapter, showcasing how a robust and scalable solution approach can empower airlines to manage their complex schedules under increasing operational and economic pressures.

Bibliographical note. *The content of this chapter is based on a manuscript accepted for publication: Ghamary, Y., & Chen, M. (2024). An Operational-Level MRO Scheduling Optimization for Aircraft Type C and D Checks Using Iterative Local Search. Accepted for publication in the Journal of Quality in Maintenance Engineering.*

3.1 Introduction

The aviation industry has witnessed significant growth in recent years, leading to intensified competition among airlines and an increased demand for air travel. This surge has brought about substantial challenges in maintaining aircraft in optimal condition, with Maintenance, Repair, and Overhaul (MRO) activities playing a pivotal role in ensuring safety and efficiency. Maintenance operations account for a considerable portion of operational costs and available time, compelling airlines to seek strategies that optimize maintenance schedules without compromising quality or safety. Traditionally, MRO scheduling has relied heavily on manual planning and human expertise, often supported by basic data processing tools. However, the complexity of modern aircraft, the multitude of components requiring maintenance, and the interdependencies between tasks have rendered manual scheduling insufficient. Suboptimal scheduling can lead to increased Aircraft-on-Ground (AOG) time, unplanned maintenance interventions, flight cancellations, and overall operational disruptions, all of which have significant financial implications. Despite the critical importance of efficient MRO scheduling, there is a noticeable gap in existing methodologies, particularly concerning operational-level scheduling for comprehensive maintenance checks such as Type C and D, which are the most extensive in the industry. Existing optimization models often fail to adequately address the complexities introduced by incorporating remanufacturing options into MRO processes. While remanufacturing can enhance sustainability and cost-effectiveness, it also increases scheduling complexity due to additional tasks, resource requirements, and longer completion times. Furthermore, many models do not fully capture the dynamic nature of MRO activities, particularly when considering the probabilistic nature of component failures and the impact of current maintenance decisions on future maintenance requirements. To bridge these gaps, there is a need for advanced models that integrate these factors while providing practical solutions within acceptable computational times.

This chapter focuses on developing an effective solution methodology for the MRO scheduling optimization model introduced in the previous chapter. That model formulated a mathematical programming framework that captures the operational complexities of MRO planning, incorporating current maintenance costs, anticipated future expenses, and penalties associated with unplanned interventions. To solve this model efficiently, we propose an Iterative Local Search (ILS) heuristic designed to produce optimal or near-optimal solutions within practical

computational times. This heuristic approach is particularly well-suited for large-scale, real-world MRO applications where exact methods often become intractable. The proposed ILS is tailored to address the key challenges of the problem, including task sequencing, shared resource constraints, interdependencies among activities, and component reliability thresholds. A central innovation of this work lies in integrating remanufacturing as a viable maintenance option, enabling a more cost-effective and forward-looking scheduling strategy. This integration not only enhances model realism but also provides a more flexible decision-making framework for MRO planners. To evaluate the effectiveness of the proposed approach, computational experiments are conducted on both small and large problem instances. Results demonstrate that the heuristic delivers high-quality solutions more efficiently than conventional optimization methods. While earlier research has explored individual elements of MRO scheduling, few have addressed the dual challenge of incorporating remanufacturing and developing scalable heuristics for complex problem settings. This chapter addresses that gap, offering a practical solution tool that supports improved maintenance planning, reduced delays, and better operational reliability for airlines and MRO service providers.

In the remainder of this chapter, Section 3.2 presents a literature review, focusing on studies that incorporate operational-level MRO scheduling, remanufacturing strategies, and heuristic solution methods. Section 3.3 then describes the problem setting, outlining the scope, assumptions, and constraints that guide the modeling framework. Following that, Section 3.4 details the proposed solution methodology, highlighting the Iterative Local Search (ILS) approach and how it efficiently addresses large-scale scheduling complexities. Section 3.5 offers numerical examples, demonstrating the model's and heuristic's performance across both small-scale and large-scale scenarios. Finally, Section 3.6 concludes the chapter by summarizing key findings and discussing avenues for future enhancements and research directions.

3.2 Literature Review

This section reviews existing literature on MRO scheduling, focusing on Type C and D aircraft maintenance activities and the integration of remanufacturing options.

3.2.1 Challenges in MRO Scheduling

The aviation industry is experiencing significant growth, leading to challenges in maintaining aircraft in optimal condition (Cooper et al., 2018). This expansion increases not only the number of aircraft but also the complexity and cost of MRO operations (Qin, Zhang, et al., 2020). Airlines are continually seeking ways to minimize MRO costs while maintaining safety and quality standards (Ertogral & Öztürk, 2019). Efficient inventory management and timely services are crucial, especially for components requiring regular overhaul (Shirinfar et al., 2022). MRO scheduling is inherently complex, with maintenance constraints recognized as fundamental in aircraft scheduling (Sriram & Haghani, 2003). The unpredictability of flight arrivals, influenced by factors such as weather and mechanical failures (Song et al., 2022), further compounds this complexity. In response to these challenges, predictive maintenance strategies leveraging data-driven techniques are increasingly adopted to enhance maintenance scheduling and operational reliability (Rivas et al., 2022).

3.2.2 Optimization Models in MRO Scheduling

Various optimization techniques and mathematical programming models have been proposed for MRO scheduling. These models aim to optimize objectives such as cost, profit, or service levels while considering operational constraints. (Yan et al., 2008) introduced a binary integer programming model for long-term Aircraft Maintenance Check Scheduling (AMCS), utilizing CPLEX to identify optimal maintenance types and timings within a planning period. Their model focuses on strategic-level scheduling without incorporating remanufacturing options. (Deng et al., 2020) developed a dynamic programming method enhanced with Reinforcement Learning (RL) for extended AMCS challenges. Their approach integrates future mission requirements, repair expenses, spare part inventory, and aircraft prognostics. However, the model's high computational complexity limits its applicability to large-scale problems. (Qin, Ma, et al., 2020) concentrated on MRO procurement and formulated a scenario-based stochastic programming model to optimize total MRO cost. Their work addresses spare parts planning under uncertainty but does not consider operational-level scheduling with remanufacturing options. (Hasancebi et al., 2023) presented a fuzzy mixed-integer linear programming model for workforce optimization in MRO operations. They focused on enhancing labor productivity and airworthiness by assigning critical tasks to experienced technicians, addressing human resource allocation rather than scheduling

complexities. (G. Liu et al., 2023) developed a maintenance stand scheduling model using mixed-integer programming to address spatiotemporal constraints in MRO. While their model improves maintenance stand utilization, it does not integrate remanufacturing options into the scheduling process. (Semaan & Yehia, 2019) developed a stochastic detailed scheduling model for periodic maintenance of military rotorcraft, utilizing a cyclical operation network combined with Monte Carlo simulations. Their model addresses uncertainties in maintenance but is specific to military applications and does not consider remanufacturing. (Verhagen & De Boer, 2018) aimed to decrease the frequency of unscheduled maintenance activities by developing a predictive maintenance model using proportional hazard models. Their focus is on predicting component failures rather than integrating remanufacturing into scheduling.

3.2.3 Heuristic and Metaheuristic Methods in MRO Scheduling

Due to the NP-hard nature of MRO scheduling problems, heuristic and metaheuristic methods have been employed to find near-optimal solutions within reasonable computational times. (Weide et al., 2022) developed a robust scheduling framework based on Genetic Algorithms (GA) for extensive aircraft maintenance checks. The solution method minimizes total workload and schedule revisions under uncertainty. However, it does not include remanufacturing options. (Pimapunsri et al., 2022) also used GA to solve resource-constrained project scheduling problems in aircraft heavy maintenance, aiming to reduce overall project duration. Their approach addresses resource constraints but does not integrate remanufacturing into the scheduling model. (Torkashvanda et al., 2023) introduced a pioneering multi-stage flow shop scheduling method for equipment overhaul. This method combines GA and Particle Swarm Optimization (PSO) to efficiently tackle complex maintenance tasks. While effective for equipment overhaul, it lacks integration with remanufacturing options in the aviation context. (Qin et al., 2019) developed a mathematical model and algorithms for the aircraft hangar maintenance scheduling problem, focusing on minimizing penalties due to delays in extensive maintenance tasks. Although their model addresses scheduling efficiency, it does not incorporate remanufacturing. (Pimapunsri & Weeranant, 2018) used Design Structure Matrix (DSM) to map the complex network of activities involved in comprehensive aircraft maintenance, addressing resource constraints in scheduling. Their work improves scheduling efficiency but does not consider remanufacturing options. (Samaranayake & Kiridena, 2012) introduced an integrated framework for major aircraft

maintenance incorporating uncertainties. Their model enhances maintenance planning but does not include remanufacturing in the scheduling process. (Kennet, 1994) proposed a regenerative optimal stopping model for aircraft engine maintenance, balancing risks and costs of engine failures against overhaul expenses using stochastic dynamic programming. While valuable, this model focuses on engine maintenance without integrating remanufacturing strategies. (Antonakis & Giannakoglou, 2018) optimized engine maintenance for military aircraft using Monte Carlo simulation and PSO, considering engine part shortages. Their approach is specific to military applications and does not address remanufacturing in scheduling. (Stranjak et al., 2008) developed a multi-agent simulation system for aircraft engine overhaul scheduling. While innovative, their model does not incorporate remanufacturing options.

3.2.4 Remanufacturing in Aviation MRO

Remanufacturing plays a critical role in enhancing cost-effectiveness and environmental sustainability in MRO processes. The environmental and operational benefits of adopting remanufacturing in aircraft maintenance are highlighted by (Asmatulu et al., 2013), emphasizing ecological and efficiency gains achievable through remanufacturing. (Ferrer et al., 2011) demonstrated the pivotal role of remanufacturing in MRO operations through the application of RFID technology in large job shop remanufacturing, aligning with the aviation industry's shift towards sustainable practices as noted by (Hashemi et al., 2016). However, these studies focus on operational improvements without integrating remanufacturing into scheduling models. (Rajamani et al., 2004) provided a cost-benefit analysis of incorporating remanufacturing techniques for condition-based maintenance of aircraft engines. Similarly, (Yang & Zheng, 2016) discussed the complexities of remanufacturing, focusing on its impact on coupled components within aerospace engine MRO operations. While these studies underscore the importance of remanufacturing, they do not integrate it into scheduling frameworks. (Johnson et al., 1999) applied computer simulation in planning and scheduling aerospace engine maintenance, effectively integrating remanufacturing processes. Yet, their model is specific to engine maintenance and lacks a generalized approach for operational-level scheduling. (Zheng et al., 2019) investigated the dynamics of aircraft part remanufacturing systems, proposing an adaptive rescheduling strategy under dynamic environments. (Lee et al., 2001) discussed the intricate process of remanufacturing aircraft fuel pumps, underlining the crucial role of data and information technology. Both studies highlight the

significance of remanufacturing but do not incorporate it into comprehensive MRO scheduling models.

3.2.5 Advanced Studies and Research Gap

While significant progress has been made in MRO scheduling and remanufacturing, several advanced studies highlight the need for further integration of these areas. (Ichou, 2023) explored the influence of digitization and automation on MRO, emphasizing the need for MRO companies to embrace modern methodologies to enhance service quality. (Karaoğlu et al., 2023) provided a comprehensive review of machine learning and deep learning applications in predictive maintenance, underscoring their growing role in optimizing aircraft maintenance systems. (Song et al., 2022) proposed a self-learning state-aware rescheduling approach for robot-aided aero-engine inspections, demonstrating advancements in automated maintenance processes. (Tseremoglou et al., 2023) proposed Deep Reinforcement Learning (DRL) as an alternative to Mixed Integer Linear Programming (MILP), addressing uncertainty more effectively in fleet maintenance scheduling. Additionally, (L. Wang et al., 2024) developed a predictive maintenance scheduling framework for aircraft engines based on Remaining Useful Life (RUL) prediction, aiming to optimize engine maintenance and reduce downtime.

Despite these advancements, the integration of remanufacturing options into operational-level MRO scheduling models for Type C and D maintenance checks remains limited. Existing models often focus on strategic-level planning or address remanufacturing and scheduling separately. Moreover, there is a lack of efficient heuristic methods capable of handling the complexities of large-scale MRO scheduling problems with remanufacturing options.

3.2.6 Contribution of the Present Study

To address these gaps, this chapter used the developed mathematical programming model in the previous chapter that integrates remanufacturing options into operational-level MRO scheduling for Type C and D maintenance checks. Additionally, an efficient heuristic solution method based on Iterative Local Search (ILS) is proposed to solve the model, ensuring optimal or near-optimal solutions suitable for large-scale, real-world applications. The effectiveness and efficiency of the proposed approach are demonstrated through computational experiments on both small-scale and large-scale instances, providing significant contributions to the field of aviation MRO scheduling.

3.1.1 Authors' Contribution Table

To illustrate the novelty and contribution of the present study compared to existing literature, Table 3.1 Comparison of existing studies and present work summarizes key aspects of selected studies.

Table 3.1 Comparison of existing studies and present work

Reference	Problem Addressed	Methodology	Remanufacturing Included	Maintenance Type	Solution Approach	Gaps in Study
(Yan et al., 2008)	Long-term aircraft maintenance scheduling	Binary Integer Programming	No	Not Specified	Exact methods (CPLEX)	No remanufacturing, strategic-level focus
(Deng et al., 2020)	Maintenance check scheduling optimization	Dynamic Programming	No	Not Specified	DP enhanced with RL	High computational complexity, no remanufacturing
(Qin, Ma, et al., 2020)	Spare parts planning in MRO provider	Scenario-based Stochastic Programming	No	Not Specified	Stochastic programming	Does not address operational-level scheduling
(Hasancebi et al., 2023)	Aircraft maintenance workforce optimization	Fuzzy Mixed-Integer Linear Programming	No	Not Specified	Mathematical modeling	Focuses on workforce, not scheduling with remanufacturing
(G. Liu et al., 2023)	Maintenance stand scheduling with spatiotemporal constraints	Mixed-Integer Programming	No	Maintenance stand allocation	Mathematical modeling	Does not include remanufacturing
(Weide et al., 2022)	Robust long-term heavy maintenance scheduling	Genetic Algorithm	No	Heavy maintenance checks	Heuristic method	Does not include remanufacturing
(Pimapunsri et al., 2022)	Resource-constrained project scheduling in heavy maintenance	Genetic Algorithm	No	Heavy maintenance	Heuristic method	Does not include remanufacturing
(Torkashvanda et al., 2023)	Equipment scheduling and assignment in overhaul industry	GA and PSO	No	Overhaul industry	Metaheuristic methods	Lacks integration with remanufacturing in aviation context
(Ferrer et al., 2011)	RFID application in remanufacturing operations	RFID Technology Application	Yes	Remanufacturing processes	Operational improvements	Does not integrate remanufacturing into scheduling models

Table 3.2 Comparison of existing studies and present work (Continued)

(Rajamani et al., 2004)	Condition-based maintenance for aircraft engines	Cost-benefit Analysis	Yes	Engine maintenance	Analytical methods	Does not integrate remanufacturing into scheduling models
(Yang & Zheng, 2016)	Managing coordination complexity in engine remanufacturing	Analytical Study	Yes	Engine remanufacturing	Conceptual framework	Lacks operational-level scheduling models with remanufacturing
(Zheng et al., 2019)	Rescheduling strategy for aircraft parts remanufacturing	Adaptive Rescheduling Strategy	Yes	Aircraft parts remanufacturing	Rescheduling under dynamics	Does not integrate with operational-level scheduling
(Lee et al., 2001)	Tracing the repair of aircraft fuel pumps in remanufacturing	Information Products & Data Analysis	Yes	Aircraft fuel pumps	Data-driven methodology	Lacks integration into scheduling models
(Ichou, 2023)	Automation in Aircraft Maintenance (Maintenance 4.0)	Digitalization and Automation	No	Aircraft operational processes	Digital workflow optimization	Does not incorporate remanufacturing and heuristic scheduling
(Karaoğlu et al., 2023)	Applications of machine learning in aircraft maintenance	Machine Learning	No	Predictive maintenance	AI-driven optimization	Does not consider scheduling under operational constraints
(Tseremoglou et al., 2023)	Optimization models for aircraft fleet condition-based maintenance scheduling	Mixed-Integer Linear Programming (MILP) vs. Reinforcement Learning (RL)	No	Fleet maintenance	Comparative study (MILP vs. RL)	Lacks integration of remanufacturing and heuristic methods
(L. Wang et al., 2024)	Predictive maintenance scheduling for aircraft engines	Remaining Useful Life (RUL) Prediction & AI	No	Aircraft engines	AI-driven predictive maintenance	Focuses on RUL prediction but does not integrate scheduling optimization
(Y. Hu et al., 2021)	Long-term maintenance decision optimization	Reinforcement Learning	No	Not Specified	RL-driven strategy	Does not consider remanufacturing, focuses on long-term decisions
Present Study	Operational MRO scheduling with remanufacturing	Mathematical Modeling, ILS Heuristic	Yes	Type C and D maintenance	Heuristic method (ILS)	-

3.3 Problem Description and Modeling

This chapter addresses the same Maintenance, Repair, and Overhaul (MRO) scheduling problem introduced in Chapter 2. The model captures the complexity of heavy aircraft maintenance activities (C-checks and D-checks) and incorporates decision-making over multiple maintenance actions—namely repair, replacement, and remanufacturing. These actions differ in terms of resource requirements, cost implications, and long-term reliability effects.

The full mathematical programming model, including its assumptions, decision variables, constraints, and cost components, has already been presented in section 2.3. Readers are referred to that section for a detailed formulation of the optimization problem. To avoid redundancy, the model is not repeated here.

The focus of this chapter is on developing a computationally efficient solution method to solve the model described in Chapter 2. Specifically, an Iterative Local Search (ILS) heuristic is designed and implemented to produce high-quality solutions for both small- and large-scale instances of the problem. The performance of the proposed heuristic is evaluated against exact optimization results (where feasible), and the solution quality, runtime, and scalability are analyzed in depth. The details of the developed solution methodology are given in the next section.

3.4 Solution Methodology

In this section, we present the solution methodology developed to address the complex MRO scheduling problem described earlier. Given the NP-hard nature of the problem due to numerous integer variables and constraints, exact methods are computationally infeasible for large-scale instances. To overcome this challenge, a heuristic-based approach is proposed to efficiently obtain high-quality solutions within a reasonable computation time. The proposed heuristic framework consists of three main components. First, a constructive heuristic is employed to efficiently generate an initial feasible solution, ensuring that all constraints related to task sequencing, resource allocation, and activity dependencies are satisfied. Following this, a local search heuristic is applied to refine the initial solution by systematically exploring the solution space neighborhood, aiming to identify potential improvements. Finally, an Iterative Local Search (ILS) metaheuristic is utilized to iteratively enhance the solution by modifying task assignments and applying localized

refinements, ultimately leading to optimal or near-optimal solutions. The details of each heuristic component and the overall solution framework are elaborated in the subsequent subsections.

3.4.1 Constructive Heuristic

The constructive heuristic is designed to efficiently generate an initial feasible schedule for the MRO problem by making strategic decisions on task selection and activity sequencing. This heuristic employs a greedy approach aimed at the total expected cost for each component. This accounts for current maintenance costs, estimated future maintenance costs, and the potential expenses arising from unplanned maintenance events. It is also insured that the selected tasks for essential components meet reliability thresholds. Once the tasks are selected, the heuristic prioritizes activities based on their duration and resource utilization. By doing so, it provides a practical starting point for further optimization through local search. The constructive heuristic follows a systematic approach, progressing through five key stages:

1. **Task Evaluation and Selection:** Each component is evaluated for potential maintenance tasks based on total expected cost (TEC). Essential components are tasks are selected from the options that ensures the reliability thresholds are met.
2. **Activity Scheduling Preparation:** The algorithm identifies precedence relationships between tasks, ensuring that required dependencies are satisfied before execution.
3. **Activity Prioritization:** Activities are sorted based on Weighted Resource Usage (WRU), prioritizing those that consume fewer resources relative to their availability. In cases of identical WRU values, activities with shorter durations are given precedence.
4. **Scheduling Execution:** Activities are assigned start times while checking for resource availability at each time step. The process continues iteratively until all activities are scheduled.
5. **Finalization:** Any components requiring no action are marked as completed, and the final schedule is assessed for feasibility and efficiency.

The constructive heuristic is developed to efficiently generate feasible maintenance schedules while ensuring compliance with operational constraints and resource limitations. The heuristic strictly enforces resource constraints by ensuring that labor and machinery, which are shared across multiple maintenance tasks, are never overallocated at any given time. By maintaining resource feasibility, the heuristic prevents scheduling conflicts and ensures that maintenance

activities can be executed without unnecessary delays. Additionally, precedence constraints are incorporated to maintain logical task execution. At the component level, dependencies between tasks dictate that certain maintenance operations must be completed before others can commence. At the activity level, sequencing constraints ensure that maintenance procedures are performed in a structured and logical order, adhering to established industry protocols. A fundamental objective of the heuristic is to minimize idle time and increase resource utilization efficiency. To achieve this, a prioritization strategy is employed that ranks activities based on their Weighted Resource Usage (WRU). Activities that require fewer resources relative to their availability are prioritized, enabling an efficient allocation of limited maintenance capacity. In cases where multiple activities exhibit similar WRU values, tasks with shorter durations are scheduled first. This prioritization mechanism supports cost-effective scheduling decisions, ensuring that resource utilization remains balanced throughout the maintenance process.

Beyond feasibility and efficiency, the constructive heuristic offers several advantages in MRO scheduling optimization. By enforcing task dependencies and resource constraints, it guarantees the feasibility of the generated schedule, ensuring compliance with operational requirements. Its computational efficiency allows it to rapidly generate an initial solution, making it particularly well-suited for large-scale scheduling problems, where exact methods would be computationally impractical. Additionally, the heuristic enhances cost-effectiveness by structuring maintenance schedules in a way that reduces total expected maintenance costs, incorporating future maintenance needs and potential unplanned interventions into the decision-making process. The method is also highly scalable, making it adaptable to problems of varying sizes and complexities, thereby offering a flexible and robust solution for diverse MRO environments. By integrating scheduling constraints, resource prioritization strategies, and computational efficiency, the heuristic not only generates a practical and feasible maintenance schedule but also establishes a strong foundation for further refinement through iterative local search (ILS) techniques.

The following section presents Algorithm 3.1, which provides a detailed breakdown of the constructive heuristic, including its computational steps and logical conditions.

Step 1: Task Evaluation and Selection

- For each Component $j \in CPN$:
 - Calculate the Total Expected Cost (TEC) for each possible maintenance task:

$$TEC_{jr} = C_{jr} + ECN_{jr} + ECU_j \quad j \in CPN, r \in TSK \quad (21)$$
 - Task Selection Criteria:
 - Essential Components ($ES_j = 1$): Within the tasks where the probability of non-failure (PN_{jr}) meets the minimum reliability threshold (MR), select the one with the lowest TEC_{jr} .
 - Non-Essential Components ($ES_j = 0$): Consider all tasks, selecting the one with the lowest TEC_{jr} .

Step 2: Activity Scheduling Preparation

- Identify Precedence Relationships:
 - Determine any precedence constraints between components and their tasks. For instance, certain components may need to be serviced before others due to operational dependencies.
 - Check the precedence of activities within each selected task to ensure proper sequencing.
- Eligible Activities to Start (EAS) List:
 - Initialize the EAS list with activities that:
 - Have all predecessor activities completed or not required.
 - Have not yet started.

Step 3: Activity Prioritization

- Calculate Weighted Resource Usage (WRU_{jrk}) for each activity k of task r in component j :

$$WRU_{jrk} = \sum_{l \in RES} \frac{RN_{jrk}l}{AR_l} \quad j \in CPN, r \in TSK, k \in ACT \quad (22)$$
- Purpose of WRU :
 - The WRU_{jrk} metric prioritizes activities that consume fewer resources relative to their availability, promoting efficient resource utilization.
- Activity Sorting:
 - Primary Criterion: Sort activities in the EAS list in ascending order of WRU_{jrk} values (activities with lower WRU_{jrk} have higher priority).
 - Secondary Criterion: In case of equal WRU_{jrk} values, prioritize activities with shorter durations (D_{jrk}).

Step 4: Scheduling Loop

- Initialize Time Period $t=1$.
- While there are unscheduled activities:
 - For each activity in the *EAS* list:
 - Resource Availability Check:
 - Verify if the required resources for the activity are available at time t .
 - If resources are available:
 - Assign Start Time: Schedule the activity to start at time t .
 - Calculate Completion Time:

$$t_{Completion} = t + D_{jrk} - 1 \quad (23)$$
 - Update Resource Allocation:
 - Deduct the resources used by the activity from the available resources for the duration of the activity.
 - Move Activity to Running Activities (*RA*) List.
 - If resources are not available:
 - Proceed to the next activity in the *EAS* list.
 - If all the resources reach 0, the heuristic proceeds to the next time period without going through the full *EAS* list.
 - Advance Time: Increment t by 1.
 - Update Resource Availability and Lists:
 - Complete Activities:
 - For any activity in the *RA* list that completes at time t , remove it from the *RA* list and release its resources back to the pool.
 - Update *EAS* List:
 - Add new activities to the *EAS* list whose predecessor activities have just been completed.
 - Repeat the Scheduling Loop until all activities are scheduled

Step 5: Finalizing the Schedule

- Components with No Action Required: For components where "No Action" was selected, mark them as completed at the start of the schedule.

This initial solution, while feasible and cost-effective, can be further improved through local search procedures that iteratively refine the schedule by exploring alternative task sequences and activity allocations; the next section presents the design and implementation of the Iterative Local Search (ILS) strategy used to enhance solution quality.

3.4.2 Local Search Heuristic

The Local Search Heuristic is designed to improve the initial feasible schedule obtained from the Constructive Heuristic by exploring the solution space in the neighborhood of the current schedule. It systematically adjusts the sequence of activities to find a schedule with a lower total cost while ensuring that all constraints, such as precedence relationships and resource limitations, are satisfied.

The Local Search Heuristic operates as an iterative improvement process that refines the initial schedule obtained from the Constructive Heuristic. The algorithm systematically evaluates alternative sequencing options to identify cost-reducing modifications, ensuring that any changes maintain schedule feasibility. The process consists of four key phases:

1. **Assignment of Initial Ranks to Activities:** Each activity is assigned a numerical rank based on its order in the initial schedule. The earliest scheduled activity is assigned Rank 1, followed by the next activity with Rank 2, and so forth. These ranks define the relative scheduling order of activities during the optimization process.
2. **Exploration of Neighboring Solutions:** The algorithm systematically iterates over activities to evaluate potential swaps that could lead to an improved schedule. For each activity, a set of neighboring activities is identified based on precedence compatibility, component independence, and rank differences. If a valid neighbor is found, the algorithm swaps their ranks and reschedules the activities accordingly.
3. **Cost Evaluation and Decision Process:** The total cost of the revised schedule is computed. If the modified schedule results in a lower total cost, the change is accepted, and the algorithm proceeds to explore additional improvements in the updated neighborhood. If the new schedule does not yield cost reductions, the swap is reversed, and the previous configuration is restored.
4. **Termination and Output of the Improved Schedule:** The process continues until a full iteration through all activities results in no accepted swaps, indicating that no further

improvements can be made. At this point, the algorithm terminates, and the best-discovered schedule is finalized. The final schedule is recorded, including start and finish times for all activities and the updated total cost.

To illustrate the detailed computational steps and logic, the implementation of the local search heuristic, including key equations and pseudocode, is presented in Algorithm 3.3.

Algorithm 3.3 Local search heuristic

Step 1: Assign Initial Ranks to Activities

- a. Initialize Activity Ranks
 - i. Assign Ranks Based on Initial Schedule:
 - For each activity in the initial schedule, assign a rank corresponding to its scheduled order.
 - The earliest scheduled activity is assigned Rank 1, the next activity Rank 2, and so on.

Step 2: Iterate Over Activities to Explore Neighborhoods

- a. For Each Activity i in the Activity List:
 - i. Identify Neighboring Activities
 - Criteria for a Neighboring Activity j :
 - Precedence Compatibility: Activity j is not a predecessor or successor of activity i .
 - Component Independence: The components associated with activities i and j are not directly dependent (no direct precedence relationship between their components).
 - Rank Condition: Activity j has a higher rank than activity i , i.e., $Rank_j > Rank_i$.
 - Different Start Times: Activities i and j do not start at the same time in the current schedule.
 - ii. For Each Neighboring Activity j :
 - Swap Ranks of Activities i and j
 - Exchange Ranks:
 - Update the ranks as follows:

$$Rank_i^{New} = Rank_j^{Old}$$

$$Rank_j^{New} = Rank_i^{Old}$$
 - This effectively swaps the positions of activities i and j in the scheduling order.
 - Reschedule Activities Based on New Ranks
 - Order of Scheduling:

Algorithm 3.4 Local search heuristic (Continued)

- Reschedule all activities in ascending order of their new ranks.
 - Activities with lower ranks are scheduled before those with higher ranks.
- Constraint Verification:
 - Precedence Constraints:
 - a. Ensure that all activity and component precedence relationships are maintained.
 - b. An activity cannot start before its predecessors are completed.
 - Resource Constraints:
 - a. Verify that resource usage at any time does not exceed available resources.
 - Activity Sequencing:
 - a. Maintain the correct sequence of activities within each task.
- Calculate Total Cost of the New Schedule
 - Compute Total Cost:
 - Sum the following cost components:
 - a. Current Maintenance Costs (C_{jr}): Costs of performing tasks in the current maintenance window.
 - b. Estimated Future Maintenance Costs (ECN_{jr}): Anticipated costs in the next maintenance window.
 - c. Estimated Unplanned Maintenance Costs (ECU_j): Expected costs due to potential unplanned maintenance.
 - d. Project Delay Penalties or Early Completion Rewards:
 - 1. Penalties incurred if the project exceeds the due date.
 - 2. Rewards (bonuses) received if the project is completed ahead of schedule.
- Decision Making
 - Compare Total Costs:
 - If the New Total Cost is Lower Than the Original Total Cost:
 - a. Accept the New Schedule:
 - 1. Update the current schedule to the new schedule obtained after swapping.
 - 2. Retain the swapped ranks for activities i and j .
 - b. Repeat Step 2.a.i for Activity i :
 - 1. Since activity i now has a new rank, revisit its neighborhood to explore further improvements.

Algorithm 3.5 Local search heuristic (Continued)

- Else:
 - a. Reject the Swap:
 - 1. Revert to the previous schedule before the swap.
 - 2. Restore the original ranks of activities i and j .
- Proceed to the Next Neighboring Activity
 - Continue the Process:
 - Move to the next activity j in the neighborhood of activity i .
 - Repeat step 2.a.ii for each neighboring activity.
- If All Neighbors Have Been Explored for Activity i
 - Move to the Next Activity:
 - Proceed to the next activity $i+1$ in the activity list.
 - Repeat steps 2 for the new activity.

Step 3: Termination Condition

- a. Repeat the Process
 - i. Continue Iterating:
 - Repeat Step 2 for all activities in the activity list.
 - The process continues until a full pass through the activity list results in no accepted swaps (i.e., no further improvements in total cost).
- b. End of Local Search
 - i. Termination:
 - When no better schedule can be found through rank swapping after a complete iteration, the Local Search Heuristic terminates.

Step 4: Output the Improved Schedule

- a. Finalize the Schedule
 - i. Best Schedule Found:
 - The schedule with the lowest total cost obtained during the local search is considered the improved schedule.
 - ii. Record Details:
 - Document the start and finish times of all activities.
 - Note the updated total cost of the schedule.

By systematically exploring alternative sequencing options, the Local Search Heuristic provides several benefits in MRO scheduling optimization. The approach improves schedule quality by finding more cost-effective activity sequences, reducing unnecessary downtime and optimizing resource usage. Since feasibility constraints are strictly enforced, the heuristic ensures that the resulting schedule remains practical and implementable. The method is also computationally efficient, as it focuses on promising areas of the solution space, reducing the computational burden associated with exhaustive search methods. Additionally, the heuristic is adaptable, allowing for modifications that prioritize specific components or activities based on operational requirements.

This improved solution then serves as a strong input for the Iterative Local Search (ILS) metaheuristic, which further explores the solution space through controlled perturbations and local refinements. The next section introduces this ILS framework and explains how it builds upon the strengths of both the constructive and local search heuristics.

3.4.3 Iterative Local Search

To further enhance the schedule obtained from the Local Search Heuristic, we apply an Iterative Local Search (ILS) Metaheuristic. This method iteratively explores the solution space by modifying task assignments for components and applying the constructive and local search heuristics to each new configuration. The goal is to escape local optima and find a globally better solution.

The Iterative Local Search (ILS) Metaheuristic follows a structured iterative improvement process consisting of four main phases:

1. **Initialization of the Best-Found Solution:** The process begins with the schedule obtained from the Local Search Heuristic, which is recorded as the current best solution. The corresponding total cost and task assignments for all components are stored as a reference for comparison.
2. **Exploration of Alternative Task Assignments:** Each component is systematically evaluated to determine alternative maintenance tasks that could yield a better schedule. For essential components, only tasks that meet the minimum reliability threshold (MR) are

considered. For non-essential components, all available maintenance options except the current one are explored. For each alternative task assignment, the Constructive Heuristic is applied to generate a new feasible schedule, followed by the Local Search Heuristic to further refine it.

3. **Cost Evaluation and Acceptance Criteria:** The total cost of the new schedule is computed. If the new total cost is lower than the current best cost, the new schedule is accepted, and the process moves to the next component. Otherwise, the task assignment is reverted to its original state, and the search continues with the next alternative task.
4. **Termination and Output of the Best-Found Schedule:** The process iterates until no further improvements can be made. The algorithm terminates when one of the stopping criteria is met: no improvements are found after a complete iteration, a maximum number of iterations is reached, or a predefined time limit is exceeded. The best-discovered schedule is then finalized, documenting the task assignments, activity timelines, and total cost.

To illustrate the detailed computational steps and logic, the implementation of the ILS heuristic, including key equations and pseudocode, is presented in Algorithm 3.6.

Algorithm 3.6 Iterative Local Search (ILS) Heuristic

Step 1: Initialize the Best-Found Solution

- a. Set the Initial Solution
 - i. Use the schedule obtained from the Local Search Heuristic as the initial solution.
 - ii. Record the total cost of this solution as the current best cost.
 - iii. Set the current task assignments for all components.

Step 2: Iterate Over Components to Explore Alternative Tasks

For Each Component j in the Component List

- a. Generate a List of Alternative Tasks
 - Essential Components ($ES_j = 1$):
 - Include alternative tasks that meet the minimum reliability threshold (MR).

Algorithm 3.7 Iterative Local Search (ILS) Heuristic (Continued)

- Non-Essential Components ($ES_j = 0$):
 - Include all available tasks (maintenance options).
- Exclude the current task assigned to component j .
- b. For Each Alternative Task r' in the List
 - i. Modify Task Assignment for Component j
 - Assign Alternative Task r' :
 - Update the task assignment for component j from the current task r to the alternative task r' .
 - ii. Apply Constructive Heuristic
 - Generate a New Schedule:
 - Use the Constructive Heuristic to create a feasible schedule based on the new set of task assignments.
 - iii. Apply Local Search Heuristic
 - Improve the New Schedule:
 - Apply the Local Search Heuristic to the newly generated schedule to further optimize it.
 - iv. Calculate Total Cost of the New Schedule
 - Compute Total Cost:
 - Sum the current maintenance costs, estimated future maintenance costs, estimated unplanned maintenance costs, and any penalties or rewards associated with project completion time.
 - v. Decision Making
 - Compare Total Costs:
 - If the New Total Cost is Lower Than the Current Best Cost:
 - Accept the New Solution:
 - a. Update the current best solution to the new schedule.
 - b. Update the current best cost to the new total cost.
 - c. Note: An improvement has been found.
 - Break Inner Loop:
 - a. Since an improved solution was found for component j , proceed to the next component.
 - Else:
 - Reject the New Task Assignment:
 - a. Revert the task assignment for component j back to the original task r .
 - vi. Proceed to the Next Alternative Task
 - Continue the process with the next alternative task r' for component j .

Algorithm 3.8 Iterative Local Search (ILS) Heuristic (Continued)

- c. If All Alternative Tasks Have Been Explored for Component j
 - i. Proceed to the Next Component:
 - Move to the next component $j+1$ in the component list.

Step 3: Termination Condition

- a. Repeat the Process
 - i. Continue Iterating:
 - Repeat Step 2 for all components in the component list.
 - ii. Convergence Criteria:
 - The algorithm can terminate based on one of the following conditions:
 - No Improvement Found:
 - After a complete pass through all components, no better solution is found.
 - Maximum Iterations Reached:
 - A predefined number of iterations have been completed.
 - Time Limit Exceeded:
 - The algorithm has run for a predetermined maximum computational time.

Step 4: Output the Best-Found Schedule

- a. Finalize the Schedule
 - i. Best Schedule Found:
 - The current best solution is considered the final schedule.
 - ii. Record Details:
 - Document the task assignments for all components.
 - Record the start and finish times of all activities.

Note the total cost of the best schedule.

The ILS Metaheuristic incorporates several considerations to maintain feasibility and efficiency throughout the iterative search process. The modification of task assignments allows the algorithm to explore alternative maintenance options, which can potentially lead to schedules with lower total costs. However, these modifications must adhere to strict reliability constraints, ensuring that all essential components satisfy the minimum reliability threshold (MR). To maintain feasibility, the Constructive Heuristic and Local Search Heuristic are reapplied following any task assignment changes, ensuring that the updated schedule remains valid in terms of resource constraints, precedence relationships, and activity dependencies. By integrating these heuristics, the ILS Metaheuristic builds upon the strengths of previous optimization steps, enabling a systematic and structured exploration of the solution space.

The ILS Metaheuristic operates under several assumptions to ensure consistency in cost evaluations and scheduling decisions. It assumes that costs, durations, and resource requirements remain constant throughout the execution process, ensuring that task modifications are evaluated under stable conditions. Additionally, the algorithm is designed for a deterministic environment, where all input data is known and does not change dynamically. While the method is effective in improving solutions, it has some limitations. The increased exploration of task assignments requires additional computational effort compared to previous heuristics, which may impact performance in extremely large problem instances. Furthermore, while the algorithm is designed to escape local optima, there remains a possibility that it may converge to a suboptimal solution if the search space is not sufficiently explored.

3.5 Numerical Examples

In this section, we present numerical examples to illustrate the developed model and demonstrate the effectiveness and efficiency of the proposed solution methodology. We first introduce a small-sized problem to showcase the application of the model and obtain an optimal solution for benchmarking purposes. Then, we present a larger-sized problem to examine the scalability and practicality of our solution methodology in real-world scenarios.

3.5.1 Small-Sized Problem

The small-sized problem involves an MRO process with 7 components, selected from a predictive maintenance study by (Verhagen & De Boer, 2018) on aircraft components. This is the same dataset used in Chapter 2. We focus on components with more than 500 maintenance events, indicating higher susceptibility to unscheduled maintenance. These components exhibited deviations from the global fleet's average unscheduled removal rate (URR), suggesting they are more prone to unplanned maintenance actions. The interrelationships among components are illustrated in Figure 2.1 of Chapter 2, which presents the network of interdependencies. The start and end nodes represent dummy components with zero duration, cost, and resource requirements. The number of activities for each component varies based on the type of maintenance task selected. As an example, the interconnections among activities for different tasks related to Component 2 were previously illustrated in Figure 2.2 of Chapter 2, along with detailed explanations.

Each aircraft component is assigned a status level ranging from 1 to 5, where Level 1 represents the best condition, and Level 5 denotes the worst condition. These status levels are determined based on post-inspection evaluations, reflecting the estimated probability of failure and overall operational risk. Maintenance actions influence component status levels in a predictable manner. Replacement or remanufacturing restores a component to Level 1, ensuring that it functions as if it were new. In contrast, repair operations improve a component's condition by one level, meaning that a component initially classified at Level 4 would be upgraded to Level 3 following a successful repair. Components are identified as essential based on their censoring rate. If a component has a censoring rate of 90% or higher, it is tagged as essential, requiring a zero-fault operation. The censoring rates are obtained from (Verhagen & De Boer, 2018).

The optimization model incorporates several cost parameters that play a crucial role in evaluating maintenance decisions and scheduling outcomes. These cost elements include flight delay costs, unplanned maintenance costs, penalty and bonus structures, and resource availability constraints, each of which is critical in shaping the scheduling process.

Flight delay costs are estimated at \$77.27 per minute for an Airbus A330 with 345 seats, aligning closely with the \$78 per minute estimate reported in the study by (Cook et al., 2004). These costs reflect the economic impact of schedule deviations, reinforcing the importance of minimizing

maintenance-induced delays. In cases of unplanned maintenance, replacement is the only viable option. Unlike planned maintenance activities, which offer flexibility in selecting between repair, remanufacturing, or replacement, unplanned maintenance requires immediate corrective action, making replacement the most practical and efficient strategy to minimize operational disruptions. To encourage timely project completion, a penalty and bonus structure is implemented. A penalty of \$500 per unit of time is incurred for delays beyond the scheduled completion time, whereas an equal bonus of \$500 per unit of time is awarded for early project completion. This cost mechanism ensures that the scheduling algorithm actively prioritizes efficiency while maintaining feasibility. Resource availability is also a key constraint in the scheduling model. The project operates under a fixed capacity of four units of labor and machinery, available throughout the project duration. This constraint necessitates strategic resource allocation to ensure that maintenance activities are scheduled efficiently without exceeding available workforce and machine capabilities.

For detailed information on the sample data used in this study, the reader is referred to Table 2.2 through Table 2.6 in Chapter 2. An overview of the data presented in these tables is briefly summarized here to support the interpretation of the numerical results in the current chapter. Table 2.3 provides comprehensive details for each component, including unit price, initial condition, censoring rate, essentiality, and the estimated cost of unplanned maintenance. Table 2.4 outlines the data required to calculate future maintenance costs, listing the possible maintenance tasks for each component along with the associated post-maintenance status probabilities. Table 2.5 presents both current and anticipated future maintenance costs, based on the selected maintenance task in the present maintenance window.

The integer programming model developed and introduced in Chapter 2—along with the problem structure and input data illustrated in Figure 2.2 and Table 2.2 to Table 2.6—was solved to optimality using the commercial solver CPLEX. The same problem instance was also addressed using the proposed heuristic algorithms developed in this chapter. The results obtained by the exact and heuristic methods are compared and discussed in the following sections.

3.5.2 Solutions of the Small-Sized Problem and Analysis

In this section, we present and analyze the solutions obtained for the small-sized problem introduced in the previous section. We compare the optimal solution obtained using the CPLEX

optimization software with the solutions generated by our heuristic methods: the Constructive Heuristic, the Local Search Heuristic, and the Iterative Local Search (ILS) Metaheuristic. The aim is to demonstrate the effectiveness of our heuristics in approaching the optimal solution while highlighting their computational efficiency.

Optimal Solution from CPLEX

The optimal solution obtained using CPLEX is summarized in Table 3.3. This table details the tasks assigned to each component, the activities within those tasks, and their corresponding start and finish times. The total minimized cost and the project completion time are also provided.

Table 3.3 Optimal solution from CPLEX

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	35	47
3 -1,152,106–3: DC starter generator	Repair	Activity 1	1	4
		Activity 2	48	68
		Activity 3	70	90
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	5	39
		Activity 3	40	90
5- 3–1574: NLG wheel & tire assembly	No Action	-	90	90
6- 728,809–1: Thermal actuator	Replace	Activity 1	6	13
7- Dummy Finish	No Action	-	90	90
Total Cost	\$166,262.75			

The maintenance schedule generated by CPLEX ensures that all maintenance tasks are assigned in a manner that balances cost efficiency, resource availability, and operational feasibility. Component 2 undergoes replacement between days 35 and 47, indicating that the condition of this component necessitated a full replacement to restore its operational reliability. Component 3 is scheduled for repair, with its associated maintenance activities spread across multiple time periods, reflecting the phased nature of the repair process. Component 4 is assigned remanufacturing, with activities distributed between the early and later stages of the project, demonstrating the extensive work required for this maintenance task. The optimization results indicate that Component 5 does not require maintenance, suggesting that its current condition is sufficient to meet operational reliability standards. Component 6, however, is scheduled for replacement early in the project, between days 6 and 13, ensuring that any potential failure risks are mitigated as early as possible.

The resulting solution achieves a total cost of \$166,262.75 and a project completion time of 90 days.

Constructive Heuristic Solution

The Constructive Heuristic generates an initial feasible schedule by selecting tasks based on the lowest Total Expected Cost (TEC) for each component. The solution generated by the Constructive Heuristic is summarized in Table 3.4.

Table 3.4 Constructive heuristic solution

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Remanufacturing	Activity 1	1	3
		Activity 2	117	192
		Activity 3	27	59
		Activity 4	3	17
		Activity 5	192	214
		Activity 6	17	27
		Activity 7	214	261
3 -1,152,106–3: DC starter generator	Remanufacturing	Activity 1	1	4
		Activity 2	4	12
		Activity 3	8	28
		Activity 4	28	48
		Activity 5	59	87
		Activity 6	87	117
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	214	248
		Activity 3	248	298
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	298	303
		Activity 2	303	338
6- 728,809–1: Thermal actuator	Replace	Activity 1	1	8
7- Dummy Finish	No Action	-	338	338
Total Cost	\$237,918.74			

The Constructive Heuristic selects remanufacturing for Component 2, leading to an extensive maintenance process with multiple activities spread over the project duration. As a result, the project completion time extends to 338 days, significantly longer than the optimal schedule obtained through CPLEX. The total cost of the schedule is \$237,918.74, which represents a substantial gap from the optimal cost due to longer project duration and higher maintenance costs.

Local Search Heuristic Improvement

The Local Search Heuristic refines the Constructive Heuristic solution by adjusting the sequencing of activities, minimizing idle times, and improving resource utilization. This adjustment leads to a more efficient schedule, as shown in Table 3.5.

Table 3.5 Local search heuristic solution

Components ⁷	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Remanufacturing	Activity 1	1	3
		Activity 2	5	80
		Activity 3	80	112
		Activity 4	8	22
		Activity 5	166	188
		Activity 6	22	32
		Activity 7	188	235
3 -1,152,106–3: DC starter generator	Remanufacturing	Activity 1	1	4
		Activity 2	32	40
		Activity 3	80	100
		Activity 4	146	166
		Activity 5	201	229
		Activity 6	235	265
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	112	146
		Activity 3	146	196
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	196	201
		Activity 2	229	264
6- 728,809–1: Thermal actuator	Replace	Activity 1	1	8
7- Dummy Finish	No Action	-	265	265
Total Cost	\$201,418.74			

Through improved sequencing and task ordering, the Local Search Heuristic significantly reduces the project completion time from 338 days to 265 days. Additionally, the total cost is lowered to \$201,418.74, narrowing the cost gap relative to the CPLEX optimal solution.

Iterative Local Search (ILS) Metaheuristic Enhancement

The ILS Metaheuristic further improves the schedule by exploring alternative task assignments and iteratively applying Constructive and Local Search Heuristics. The refined solution is presented in Table 3.6.

Table 3.6 Iterative Local Search (ILS) Solution

Components	Task	Activity	Start	Finish
1- Dummy Start	No Action	-	1	1
2- 697,071,003: Blade assembly and bearing	Replace	Activity 1	1	13
3 -1,152,106–3: DC starter generator	Remanufacturing	Activity 1	1	4
		Activity 2	62	70
		Activity 3	42	62
		Activity 4	70	90
		Activity 5	92	120
		Activity 6	120	150
4- 3–1573–1: MLG wheel & tire assembly	Remanufacturing	Activity 1	1	5
		Activity 2	8	42
		Activity 3	42	92
5- 3–1574: NLG wheel & tire assembly	Repair	Activity 1	92	97
		Activity 2	120	155
6- 728,809–1: Thermal actuator	Replace	Activity 1	1	8
7- Dummy Finish	No Action	-	155	155
Total Cost	\$167,237.35			

The ILS Metaheuristic results in a substantial improvement, reducing the total project duration to 155 days while achieving a total cost of \$167,237.35, which is very close to the optimal CPLEX solution.

Comparative Analysis of Results

A comparative evaluation of the CPLEX optimal solution and the heuristic-based approaches is presented in Table 3.7, highlighting differences in total costs, project completion times, computational performance, and cost gaps relative to the optimal solution.

Table 3.7 Comparative results and computational performance

Scenario	Cplex Solution	Constructive Heuristic	Local Search	ILS
Current Maintenance Cost	\$98,149	\$101,492	\$101,492	\$122,311
Future Maintenance Cost	\$37,237	\$3,571	\$3,571	\$3,571
Unexpected Maintenance Cost	\$27,875	\$5,854	\$5,854	\$5,854
Bonus	\$0	\$0	\$0	\$0
Penalty	\$3,000	\$127,000	\$90,500	\$35,500
Total Cost	\$166,262	\$237,918	\$201,418	\$167,237
Project End (Days)	90	338	265	155
Run Time (Seconds)	33	0.01	0.53	1.28
GAP	0%	43%	21.1%	0.58%

The results indicate that heuristic methods offer a significant reduction in computational time, with ILS achieving near-optimal results in only 1.28 seconds, compared to 33 seconds for the exact CPLEX solution. The Constructive Heuristic, while efficient in generating an initial feasible

solution within 0.01 seconds, results in a 43% cost gap relative to the optimal solution due to suboptimal task assignments and extended project duration (338 days). The Local Search Heuristic enhances this solution by reducing idle times and optimizing resource usage, shortening the completion time to 265 days and reducing the cost gap to 21.1%. The ILS Metaheuristic, through iterative task reassignment and rescheduling, further refines the solution, yielding a total cost of \$167,237, which is within 0.58% of the optimal solution while completing the project in 155 days. The comparison highlights the trade-off between computational efficiency and solution quality. While CPLEX provides an exact optimal solution, its computational time is significantly higher than that of heuristic approaches. The ILS Metaheuristic emerges as the most effective alternative, achieving a near-optimal solution in a fraction of the time required by CPLEX, demonstrating its suitability for large-scale, computationally complex MRO scheduling problems.

3.5.3 Computational Results of Larger-Sized Instances

To evaluate the effectiveness and scalability of the developed heuristic methods in practical settings, we tested them using a larger-sized example problem that closely resembles real-life MRO projects. This problem serves as a case study to demonstrate the applicability of our findings and to provide managerial insights into the practical benefits of our methodology.

The large-scale problem was structured based on a case from the Project Scheduling Problem Library (PSPLIB), incorporating 32 components (jobs). The necessary data—including activity counts, task interdependencies, durations, resource requirements, costs, and component status classifications—were systematically generated using Python under the following conditions:

- Each component had four available maintenance task options: No Action, Replacement, Repair, and Remanufacturing.
- The structure of task activities varied according to maintenance complexity. Replacement involves a single activity, repair consisting of a moderate number of activities, and remanufacturing requiring a greater number of activities due to its extensive nature. Each task contained a maximum of 10 activities.
- The network of interdependencies was established to reflect real-world maintenance constraints.

- Cost considerations were integrated to reflect higher expenses for resource-intensive maintenance tasks.

The resulting MRO project network consisted of 32 components, including dummy Start and Finish nodes, with a total of 451 activities and a highly interdependent task structure. This complex scheduling environment provided a comprehensive testing ground for evaluating the performance of the heuristic-based solution methodology.

Computational Results and Performance Analysis

The Constructive Heuristic, Local Search Heuristic, and Iterative Local Search (ILS) Metaheuristic were applied to the large-scale problem to evaluate solution quality, project completion times, and computational efficiency. The results are summarized in Table 3.8, demonstrating the relative effectiveness of each approach.

Table 3.8 Results for solution methodology on large-sized problem

Stage	Constructive	Local Search	ILS
Total Cost	\$1,061,188	\$971,188	\$787,232
Project End (Days)	1523	1343	400
Run Time (Seconds)	0.77	1146	60585
Improvement	-	8.48%	25.88%

The Constructive Heuristic quickly generated a feasible schedule, producing a total cost of \$1,061,188 and a project completion time of 1,523 days. The Local Search Heuristic significantly improved upon this result by optimizing task sequencing and resource utilization, reducing the total cost by 8.48% to \$971,188 while shortening the project duration to 1,343 days. The ILS Metaheuristic achieved the most substantial improvement, lowering the total cost by 25.88% to \$787,232 and reducing the project duration to 400 days. However, the ILS Metaheuristic required 60,585 seconds (16.8 hours) of computational time, presenting a significant computational burden despite its superior solution quality. The results confirm that the proposed heuristics effectively address complex MRO scheduling problems. The observed cost reductions and project duration improvements demonstrate the practical advantages of employing heuristic-based scheduling over traditional manual approaches. However, the extensive computational time required by the ILS Metaheuristic highlights the trade-off between solution quality and computational efficiency, particularly in time-sensitive maintenance environments. While the ILS Metaheuristic provided

the best cost reduction and schedule efficiency, its computational intensity may limit its practicality in real-world MRO operations. This highlights the need for balancing optimization depth with computational constraints, especially in industries requiring rapid decision-making.

3.5.4 Enhancing Computational Efficiency

To address the computational efficiency challenges, we conducted further experiments by modifying the local search process. Specifically, we limited the number of neighboring solutions explored in each iteration to 6, aiming to reduce computational time without significantly sacrificing solution quality. The results obtained using the limited neighborhood approach are presented in Table 3.9.

Table 3.9 Results with limited neighborhood size (6 neighbors)

Stage	Constructive	Local Search	ILS
Total Cost	\$1,061,188	\$987,688	\$806,747
Project End (Days)	1523	1376	506
Run Time (Seconds)	0.77	100.71	3597.11
Improvement	-	6.93%	23.98%

The results indicate that while limiting the neighborhood size leads to a slightly higher total cost, the computational time is significantly reduced. The Local Search Heuristic achieved an improvement of 6.93%, reducing the total cost to \$987,688 and the project duration to 1,376 days, with a runtime of just 100.71 seconds. The ILS Metaheuristic still achieved substantial cost reductions, lowering the total cost to \$806,747 while shortening the project duration to 506 days, but with a notably reduced runtime of 3,597.11 seconds (approximately one hour). The results confirm that a practical balance between solution quality and computational efficiency can be achieved by limiting the search space during optimization. The negligible loss in solution quality is offset by substantial reductions in computational time, making the approach more suitable for real-world MRO applications. By adjusting the neighborhood size in Local Search and ILS, organizations can customize the level of optimization effort based on operational constraints and urgency. This ensures that scheduling solutions remain both computationally feasible and cost-effective.

3.5.5 Practical Implications for MRO Operations

The findings from the large-scale problem confirm that the proposed heuristics offer substantial cost savings and operational efficiency when applied to MRO scheduling problems. The methodology provides key practical benefits, including:

- **Cost Savings:** The heuristics achieve significant reductions in total maintenance costs, improving overall financial efficiency.
- **Time Efficiency:** The scheduling algorithms reduce project durations, minimizing aircraft downtime and enhancing fleet availability.
- **Resource Optimization:** Efficient scheduling ensures optimal utilization of labor and machinery, preventing bottlenecks and reducing idle time.
- **Scalability:** The proposed methodology is capable of handling large-scale MRO problems, making it adaptable to varied maintenance environments.
- **Decision Support:** The insights provided by the heuristic-based scheduling framework allow managers to prioritize maintenance actions based on cost-benefit analyses, enabling informed decision-making in dynamic operational settings.

The computational results for the larger-sized problem confirm that the developed heuristics are effective and practical for solving complex MRO scheduling problems. By adjusting the local search process to limit the neighborhood size, we achieve a practical balance between solution quality and computational efficiency. This makes the methodology suitable for implementation in actual MRO operations, providing significant benefits in terms of cost savings, time efficiency, and resource optimization.

3.6 Conclusion

This chapter presented an optimization model and heuristic solution methodologies for addressing complex Maintenance, Repair, and Overhaul (MRO) scheduling problems in the aviation industry, with a key focus on incorporating remanufacturing into the scheduling process. The model aims to minimize the total maintenance cost by considering current operations, anticipated future maintenance, and potential unexpected repairs, all while adhering to operational constraints such as resource limitations and component reliability requirements.

Novel Findings

A significant contribution of this research is the integration of remanufacturing as a maintenance option within the MRO scheduling model. By including remanufacturing alongside traditional options like repair and replacement, the model captures a broader range of maintenance strategies. This integration allows for more flexible and cost-effective scheduling decisions, potentially extending the life of components, reducing waste, and promoting sustainability.

We developed efficient heuristics—the Local Search Heuristic and the Iterative Local Search (ILS) Metaheuristic—that effectively generate near-optimal solutions with significantly reduced computational time compared to traditional optimization software like CPLEX. Computational results demonstrated that incorporating remanufacturing into the scheduling process, combined with the ILS Metaheuristic, can achieve cost reductions of up to 25% in large-scale problems, highlighting its scalability and practical applicability.

Limitations

Despite the promising results, limitations exist. The heuristics may require substantial computational time for very large problems, especially when exploring extensive neighborhoods. Additionally, the model assumes deterministic input data, not accounting for uncertainties such as variable repair times or unexpected resource constraints that occur in real-world operations.

Final Remarks

The incorporation of remanufacturing into the MRO scheduling optimization model represents a significant advancement in maintenance planning. By considering remanufacturing as a viable maintenance action, organizations can achieve substantial cost savings, extend the lifespan of critical components, and contribute to environmental sustainability through reduced waste and resource consumption.

The proposed optimization model and heuristics offer practical tools for enhancing MRO scheduling efficiency. By balancing solution quality with computational efficiency, and by integrating remanufacturing options, the methodologies have the potential to significantly impact maintenance scheduling practices not only in aviation but also in other industries facing similar challenges. The demonstrated efficacy in both small-scale and large-scale problems provides a

strong foundation for the potential adoption and further development of these methods in the pursuit of improved operational efficiency and sustainability in MRO operations.

Chapter 4

Lagrangian Relaxation and Subgradient Optimization for Large-Scale MRO Scheduling

This chapter presents a Lagrangian relaxation framework to address the computational challenges of the aircraft Maintenance, Repair, and Overhaul (MRO) scheduling problem introduced in Chapter 2 and previously solved using an Iterative Local Search (ILS) heuristic in Chapter 3. The core problem integrates standard maintenance actions with remanufacturing options and includes complex constraints related to task sequencing, resource allocation, and component dependencies. While the ILS heuristic in Chapter 3 provides high-quality solutions for large instances, it does not offer lower bounds for assessing optimality. To complement that work, this chapter focuses on dual optimization by relaxing selected resource and precedence constraints using a Lagrangian relaxation approach. The resulting dual problem is solved via a projected subgradient method. Several relaxation configurations are explored to evaluate their impact on dual bound quality and computational tractability. Two adaptive step-size update rules—one progressively reducing step size, and another inspired by nonsmooth optimization techniques—are implemented to enhance convergence stability. Computational experiments on the same predictive maintenance dataset used in earlier chapters show that the proposed approach delivers strong lower bounds while remaining scalable. This allows for more rigorous performance assessment of heuristic solutions, bridging the gap between practical implementation and theoretical optimality in large-scale MRO scheduling.

***Bibliographical note.** The content of this chapter is based on a manuscript in preparation: Ghamary, Y., & Chen, M. (2024). *A Constraint Relaxation Approach for MRO Scheduling Optimization Using Lagrangian Relaxation and Subgradient Methods*.*

4.1 Introduction

In this chapter, we investigate the use of Lagrangian Relaxation techniques to tackle the large-scale aviation MRO scheduling problem introduced earlier. Given the NP-hard nature of the problem, obtaining high-quality *lower bounds* on the optimal solution is crucial for evaluating solution quality and understanding problem complexity. Lagrangian relaxation provides a way to derive such bounds by relaxing select constraints and incorporating them into the objective with penalty multipliers. By doing so, the problem structure simplifies, enabling decomposition into more tractable subproblems. Here, we leverage Lagrangian relaxation to gain insights into which constraints contribute most to computational complexity and how much cost optimality is sacrificed when they are violated. The objectives of this chapter are threefold: (1) to formulate Lagrangian-relaxed versions of the MRO scheduling model that yield computable lower bounds, (2) to apply a subgradient optimization method to iteratively adjust the Lagrange multipliers and maximize these bounds, and (3) to compare the outcomes of different relaxation strategies in terms of bound strength, convergence behavior, and solution quality. We also relate these lower bounds to the feasible solutions obtained by the previously developed Iterative Local Search (ILS) heuristic (from Chapter 3), thereby assessing how close the heuristic solutions are to optimal.

The chapter first presents results on a small problem instance, using seven different constraint relaxation scenarios to analyze gap quality and runtime. A detailed review of the convergence behavior is included to evaluate how stable or erratic the gap evolution is over iterations. Based on these results, the most promising scenario—relaxing both task dependency and labor constraints—is then selected for further testing with adaptive step-size controls. The performance of different epsilon strategies is assessed to determine whether stability and final solution quality can be improved.

Finally, the best-performing relaxation configuration is applied to the large-scale problem introduced in Chapter 3. This allows us to evaluate the scalability of the Lagrangian framework and its ability to generate useful bounds for large and complex scheduling cases. The findings provide a deeper understanding of the problem structure and support the overall evaluation of the ILS heuristic's effectiveness.

The rest of the chapter is organized as follows: Section 4.2 presents a literature review on Lagrangian relaxation methods and their applications in MRO scheduling. Section 4.3 introduces the problem description and constraint structure. Section 4.4 details the Lagrangian relaxation formulation across various constraint relaxation scenarios. Section 4.5 describes the subgradient optimization framework used to solve the resulting dual problems. Section 4.6 presents and analyzes numerical results, including experiments on small and large instances, as well as the impact of different step-size strategies. Finally, Section 4.7 concludes the chapter.

Overall, this chapter demonstrates how constraint relaxation and dual optimization can complement heuristic methods in solving large-scale MRO scheduling problems, ultimately providing a more comprehensive solution approach.

4.2 Literature Review

Mixed-Integer Linear Programming (MILP) has found wide applications across diverse industries, underscoring its societal value. Notable domains include ride-sharing cost-sharing mechanisms (S. Hu et al., 2021), home healthcare routing (Dastgoshade et al., 2021), job-shop and flow-shop scheduling (Hong et al., 2019; A. Liu et al., 2021), and freight transportation logistics (Archetti et al., 2022). However, MILP problems are inherently NP-hard due to the inclusion of integer variables, making them computationally intensive. Among the most effective strategies to address this complexity is Lagrangian relaxation, a classical technique in operations research that simplifies constrained optimization problems by dualizing selected constraints.

4.2.1 Historical Foundations and Core Concepts

One of the earliest applications of Lagrangian relaxation was by (Held & Karp, 1971), who used it to obtain strong lower bounds for the Traveling Salesman Problem through 1-tree relaxations. Their work, along with Polyak's foundational subgradient method (Polyak, 1967), demonstrated the practical viability of solving dual problems even when primal problems remain intractable. Further advances came from (Fisher & Shapiro, 1974), who laid groundwork for constructive duality in integer programming.

This technique was soon extended to real-world scheduling applications. (Muckstadt & Koenig, 1977) applied Lagrangian relaxation to power generation scheduling, illustrating how constraint

decomposition could simplify large-scale resource allocation. Similarly, (Erlenkotter, 1978) introduced a dual-based approach for the uncapacitated facility location problem, influencing many later MILP formulations.

4.2.2 Convergence and Algorithmic Enhancements

The 1990s saw growing interest in addressing the limitations of basic subgradient methods. (J.-L. Goffin & Kiwiel, 1998), and earlier (J. L. Goffin, 1977), analyzed convergence properties and rates of subgradient algorithms, identifying conditions under which they could be effective. (Kaskavelis & Caramanis, 1998) applied these ideas to large-scale job-shop scheduling, emphasizing the decomposition benefits of relaxing machine capacity constraints.

New algorithmic variants soon emerged. (Zhao et al., 1999) proposed a surrogate gradient algorithm to accelerate convergence, while (Bragin et al., 2015, 2018, 2020; Bragin & Tucker, 2022) introduced scalable and distributed surrogate Lagrangian relaxation frameworks suitable for large-scale industrial systems.

4.2.3 Expanding Applications and Sector-Specific Use Cases

(Hooker, 2024) offers a comprehensive conceptual overview of Lagrangian relaxation, outlining its use in decomposing complex optimization problems by relaxing coupling constraints (e.g., resource or time linkage). This idea has been successfully applied to multi-project scheduling (Ahmed et al., 2021), long-term production planning under uncertainty (Tolouei et al., 2021), and constrained maximization problems (Kulik et al., 2021). (Bragin, 2023) synthesizes these developments in a recent survey, noting advances in convergence, scalability, and hybridization with heuristics.

Lagrangian relaxation has also been integrated into sustainability-focused models. (Soleimani et al., 2022) developed a closed-loop supply chain framework incorporating energy efficiency, while (Sun et al., 2024) reviewed applications in workshop scheduling. (Table, 2025) extended these ideas to parallel injection machine scheduling.

4.2.4 Emerging Focus: Maintenance Scheduling and MRO

More recently, Lagrangian relaxation has begun gaining traction in the aircraft maintenance and MRO domain, though literature remains sparse. (Bulbul & Kasimbeyli, 2021) tackled the aircraft maintenance routing problem using an augmented Lagrangian and hybrid subgradient method. Their approach effectively decomposed complex scheduling constraints and enabled near-optimal routing under tight operational constraints.

Further, constraint relaxation has been explored in project scheduling with resource coupling. (Morin et al., 2016) propose relaxing resource constraints to improve tractability in resource-constrained project scheduling problems. These approaches confirm that Lagrangian relaxation is highly effective when applied to problems with tightly coupled or time-dependent constraints.

However, not all constraints are equally suited for relaxation. In job-shop settings, relaxing machine constraints simplifies subproblem independence (Kaskavelis & Caramanis, 1998), but relaxing precedence or sequencing constraints can introduce dual-space oscillation (J.-L. Goffin & Kiwiel, 1998), calling for careful constraint selection.

Despite challenges, the subgradient method remains a mainstay for optimizing the Lagrangian dual function. (Mariescu-Istodor & Fränti, 2021) show that, with proper tuning, subgradient methods can yield near-optimal dual values even in large-scale combinatorial problems.

This chapter builds on these insights from the literature by applying Lagrangian relaxation and subgradient optimization to our specific MRO scheduling model – a domain where, to our knowledge, this approach has not been extensively documented. We also contribute a comparative analysis of different constraint relaxation schemes (resources, dependencies, sequencing) within the MRO context, filling a gap in the existing research on how each type of constraint impacts solution tractability and quality.

4.3 Problem Description and Constraint Structure

Before formulating the Lagrangian relaxations, we first outline the relevant constraints of the MRO scheduling model and identify which will be relaxed. Recall from Chapter 2 that the model integrates various constraints to ensure a feasible and safe maintenance schedule. These include:

(i) task assignment constraints (each aircraft component is assigned at most one maintenance action among the options, e.g. no action, repair, replace, or remanufacture), (ii) reliability constraints (critical components must meet minimum reliability thresholds through appropriate maintenance actions), (iii) activity sequencing constraints (maintenance activities must follow logical precedence relationships and cannot overlap improperly), (iv) component dependency constraints (certain maintenance decisions on one component may require or preclude actions on another due to functional interdependencies), and (v) resource capacity constraints (limits on the availability of maintenance crews, equipment, or facilities at any given time period). All of these constraints are necessary to accurately represent the real-world scheduling problem; however, they also collectively make the model intractable to solve optimally for large instances.

In this chapter, we keep the most critical structural constraints intact – specifically, those ensuring basic feasibility and safety – and focus on relaxing a subset that are particularly burdensome computationally. The constraints we maintain as hard requirements include the one-action-per-component rule and the reliability threshold enforcement, since violating these would lead to infeasible or unsafe schedules. On the other hand, we target the following constraints for relaxation in the Lagrangian framework:

- **Resource Constraints:** These constraints enforce that at any time period, the total resource usage by all scheduled activities does not exceed the available resources (e.g. number of technicians or machines). Formally, this Constraint is of the form:

$$\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq \max(1, t-D_{jrk})}^t RN_{jrk} \times A_{jrk} \leq AR_l \quad t \in TME, l \in RES \quad (17)$$

This ensures, for example, that we do not assign more mechanics or equipment to tasks in a given hour than are actually available. The resource constraints couple the scheduling decisions across all components and time periods, making them a prime candidate for relaxation to decompose the problem.

- **Component Dependency Constraints:** These constraints capture inter-component logical dependencies in the maintenance plan. In practice, certain components' maintenance actions might depend on each other. The constraint is presented as follows:

$$\sum_{t=1}^T (t + D_{irk}) \times A_{irkt} \leq \sum_{q=1}^R \sum_{t=1}^T S_{jqt} \times t \quad i, j \in CPN \text{ with } < i, j >, r \in TSK, k \in ACT_{ir} \quad (15)$$

In general, we can express component dependency constraints as linear relationships linking decision variables of different components. They ensure the maintenance schedule is logically consistent across the entire aircraft (all components). These constraints often couple the decisions between components, preventing a simple component-by-component optimization. By relaxing them, we allow the optimization to temporarily consider infeasible combinations of component actions (e.g. treating components independently) in pursuit of a lower bound on cost.

- **Activity Sequence Constraints:** These enforce the correct precedence and sequence of activities within and across maintenance tasks. Every maintenance task (for a given component) consists of a sequence of activities that must occur in a certain order (for example, to repair a component, you must first remove it, then perform the repair, and finally re-install or test it). Additionally, some activities cannot occur in parallel due to physical or safety reasons. This constraint encompasses these ordering rules, typically ensuring that if activity A must precede activity B , then the start time of B is after the completion of A . In a linear model, such precedence can be encoded with inequalities involving start time or binary ordering variables. These sequencing constraints tie the timing of activities together and contribute significantly to the complexity of the scheduling problem, as they introduce a combinatorial ordering component. Relaxing the activity sequence constraints means the model can ignore the prescribed order or allow activities to overlap in ways that violate the original logic (for example, allowing a component's reassembly to start before its disassembly finishes in the relaxed solution). While clearly infeasible in reality, this can drastically simplify the scheduling problem structure, especially when combined with resource relaxation, by removing inter-activity timing dependencies. This constraints is presented as follows:

$$\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} \leq \sum_{t=1}^T A_{jrbt} \times t \quad j \in CPN, r \in TSK, a, b \in ACT_{jr} \text{ with } < a, b \quad (16)$$

In summary, Constraints (15), (16), and (17) will be considered for relaxation either individually or in combination. By relaxing these, we effectively remove or soften the coupling they impose: relaxing (17) decouples the schedule across time periods and resources, relaxing (15) decouples

maintenance decisions across components, and relaxing (16) decouples the internal activity ordering. The remaining constraints (such as one task per component and reliability requirements) stay in force to keep the relaxed problem bounded and meaningful. In the next sections, we formulate the Lagrangian relaxation for each case and explain how these relaxations transform the problem structure.

4.4 Lagrangian Relaxation Formulation

Lagrangian relaxation involves moving selected constraints into the objective function with associated penalty multipliers (Lagrange multipliers). By doing so, the constraints are no longer enforced strictly; instead, any violation of those constraints is penalized in the objective. The resulting problem (the Lagrangian subproblem) is easier to solve because it has fewer hard constraints, often decomposing into independent parts. However, its objective now depends on the chosen multipliers. The goal is to find the set of multipliers that maximize the objective value of the Lagrangian subproblem – this maximum is the Lagrangian dual value, which is a lower bound on the original problem's optimal cost. We formulate the Lagrangian relaxation for three sets of constraints (resource capacity, component dependency, and activity sequence) and different combination set of these constraints. For each relaxation, we present the mathematical form and discuss the intuition behind it.

3.1.2 Relaxing Resource Constraints

We first relax the resource capacity constraints (17). Let $\Lambda = \{\lambda_{l,t} \geq 0\}$ Set of non-negative Lagrange multipliers associated with each resource $l \in RES$ and time period $t \in TME$. For a given multiplier $\lambda_{l,t}$, a penalty is incurred in the objective for any usage of resource l at time t that would violate the capacity. Formally, the Lagrangian function after relaxing constraint (17) is:

$$L(\Lambda) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{t=1}^T \sum_{l=1}^L \lambda_{tl} \left(\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w=\max(1,t-D_{jr})}^t RN_{jrkl} \times A_{jrkwl} - AR_l \right) \right\} \quad (24)$$

Where:

- $\text{OriginalObjective}(x)$ is the objective function defined in the original problem in chapter 1, including maintenance costs, delay penalties, and early incentives

- The term inside the parentheses, for each pair (l, t) captures the violation of the resource constraint: total usage minus available capacity

Intuitively:

- λ_{tl} serves as a shadow price or penalty per unit of excess resource usage for resource l at time t
- If λ_{tl} is large, the optimizer is discouraged from scheduling activities that require resource l at time t
- If $\lambda_{tl} = 0$, then the corresponding constraint is effectively removed in that iteration

This relaxed formulation transforms a hard constraint into a soft penalty embedded within the objective, enabling decomposition and efficient solution strategies such as subgradient optimization. Without the resource constraints, the scheduling problem decouples by component and time. Activities no longer compete for shared resources, meaning that in the Lagrangian subproblem, each component's maintenance tasks can be scheduled as early as possible (subject only to other remaining constraints like precedence) without penalty. In fact, if resource constraints are relaxed and component dependency constraints are either relaxed or not present, then enforcing sequence constraints still allows each component's schedule to be optimized independently. This is because sequence constraints apply within individual components, and without resource or dependency-based coupling, the scheduling decisions for one component do not affect others. In such a case, the subproblem decomposes into multiple smaller problems—potentially one per component—which is significantly easier to solve than the full integrated problem. We note that any solution of this relaxed subproblem might violate resource capacities (e.g., it might schedule many resource-intensive tasks all at the same time), but its objective accounts for those violations via the λ terms. For a given set of multipliers Λ , solving $L(\Lambda)$ yields a lower-bound cost because we have allowed an infeasible (resource-violating) schedule that could be cheaper than any feasible schedule. By optimizing Λ , we seek the tightest bound, i.e. the highest cost that still underestimates the true optimum. Mathematically, the Lagrangian dual problem is given as follows:

$$\max_{\Lambda \geq 0} L(\Lambda) \tag{25}$$

Since the original MRO scheduling problem is a non-convex integer program, strong duality does not generally hold, and the optimal value of the dual $L(\Lambda)$ typically provides a lower bound rather than the exact optimal value. Nevertheless, it serves as a valuable benchmark for evaluating the quality of feasible solutions. In summary, relaxing resource constraints removes the inter-component competition for resources, yielding a subproblem that can often be solved by straightforward greedy scheduling per component. The challenge becomes finding the right multipliers Λ that balance the resource usage in the objective to produce the best possible bound. It is important to highlight that this problem is subject to the same constraint as the original problem except constraint 17 which is relaxed here. So, it is subject to constraint (9) to (16) presented in chapter 1.

4.4.1 Relaxing Component Dependency Constraints

Next, we relax the inter-component dependency constraints. Let $M = \{\mu_{ijrk} \geq 0\}$ denote the Lagrange multipliers for each constraint in set (15) – each μ_{ijrk} corresponds to a specific precedence relationship between activity k of task r for component i , and the selected task for a dependent component j .

This constraint ensures that an activity A_{irkt} , which starts at time t and lasts for duration D_{irk} , must finish before the start of any selected task q on the successor component j , whose start time is determined by the binary variable S_{jq} . We relax this dependency by incorporating it into the objective function with a penalty weighted by the corresponding multiplier μ_{ijrk} .

The Lagrangian function with component dependencies relaxed becomes:

$$L(M) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{i,j \in \langle i,j \rangle} \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \mu_{ijrk} \left(\sum_{t=1}^T (t + D_{irk}) \times A_{irkt} - \sum_{q=1}^R \sum_{t=1}^T S_{jq} \times t \right) \right\} \quad (26)$$

The term inside the parentheses represents the violation of the dependency constraint: the end time of the predecessor activity minus the start time of the dependent task

This formulation penalizes any violations where an activity finishes after the dependent task starts. By increasing the corresponding μ_{ijrk} , the optimization is discouraged from producing infeasible

solutions that break this precedence rule. This makes the relaxed model still reflect logical dependencies — though softly — and allows efficient solution through decomposition and subgradient optimization.

With the component dependency constraints (Constraint 15) removed, the maintenance planning for each component becomes significantly more decoupled — though not entirely independent due to other couplings such as shared resources (Constraint 17) and inter-component timing dependencies (Constraint 16). Ignoring Constraint 15 specifically allows the schedule to select tasks for each component without enforcing those cross-component prerequisite conditions. In effect, the aircraft is no longer treated as an interconnected system but as a collection of semi-independent components, each aiming to select its cost-optimal maintenance strategy.

The subproblem now decomposes by component: the algorithm can minimize the maintenance cost for each component individually (accounting for its task-specific costs and time penalties) and then aggregate the results. Since combinations previously prohibited by dependency constraints are now allowed — such as performing an overhaul on one component without performing a required linked task on another — the solution becomes optimistic. That is, the computed cost under these relaxed conditions may underestimate the true cost of any feasible (dependency-respecting) schedule.

The Lagrange multipliers $M = \{\mu_c\}$ serve to penalize violations of the relaxed component dependency constraints. Each μ_c corresponds to a specific dependency, and by adjusting these values during the optimization process, the objective function is reshaped to discourage infeasible solutions that break key logical relationships. High values of μ_c indicate constraints whose violation has a significant impact on feasibility or cost — effectively enforcing them through dual penalties. Conversely, low or near-zero multipliers suggest that the associated constraints are rarely binding and may be less structurally critical. Thus, the pattern of multiplier values can provide insight into the importance of specific dependencies in shaping the optimal solution.

Practically, relaxing component dependencies reveals how much cost could be saved if certain coordinated maintenance requirements were removed. If the resulting Lagrangian lower bound is close to the actual optimal cost, it suggests that these constraints do not significantly impact total

cost. Conversely, a wide gap indicates that the dependencies are critical and impose meaningful structure and cost on the problem.

The Lagrangian dual problem for these constraints is:

$$\max_{\mu \geq 0} L(M) \quad (27)$$

Solving this dual involves iteratively updating the multipliers μ to find the best balance between independent optimization and penalizing dependency violations. Still, it's important to note that Constraint (14), which governs inter-component temporal ordering, continues to couple certain components. Thus, the relaxation of Constraint (15) does not fully decompose the problem, but it simplifies it considerably.

3.1.3 Relaxing Activity Sequence Constraints

We now consider relaxing the activity sequencing constraints defined in Constraint (16). These constraints ensure that, within a maintenance task, activities follow a prescribed logical order. Let $\Gamma = \{\gamma_{j,r,a,b} \geq 0\}$ be the set of non-negative Lagrange multipliers associated with each precedence constraint between activities a and b (where $a < b$) within task r for component j . This constraint ensures that activity b does not start before activity a is completed.

To relax this constraint, we incorporate each violation as a penalty term in the objective function, weighted by the corresponding multiplier $\gamma_{j,r,a,b}$. The Lagrangian function with activity sequence constraints relaxed becomes:

$$L(\Gamma) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{j \in \text{CMP}} \sum_{r \in \text{TSK}} \sum_{(a,b) \in \text{ACT}_{jr} \text{ with } a < b} \gamma_{j,r,a,b} \times \left[\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} - \sum_{t=1}^T A_{jr bt} \times t \right] \right\} \quad (28)$$

The expression inside the parentheses represents the sequencing violation — if activity b starts before activity a finishes, the term becomes positive and is penalized. This relaxation allows for infeasible overlaps in activity scheduling (e.g., finishing installation before disassembly is complete), but penalizes such cases proportionally to $\gamma_{j,r,a,b}$. Larger values of $\gamma_{j,r,a,b}$ signal that the solver is being pushed to respect the sequencing constraint. By relaxing Constraint (16), we reduce the combinatorial complexity of the schedule, enabling more efficient computation while

still preserving the influence of sequencing through the penalty structure. The Lagrange multipliers $\gamma_{j,r,a,b}$ act as shadow prices for violating intra-task precedence, guiding the solution toward feasibility in the dual space.

When all activity sequencing constraints defined by Constraint (16) are relaxed, the time-based dependencies between activities within the same task are removed or significantly weakened. In the resulting relaxed subproblem, activities can be scheduled in any order — or even concurrently — if doing so reduces the objective value, without regard to the logical or operational sequence that would typically be enforced.

This decoupling drastically reduces the temporal complexity of the problem. For example, if resources and other constraints (like component dependencies or start-time constraints) are also relaxed or non-binding, the solver may trivially schedule all activities at time zero, resulting in a minimum makespan and minimizing any time-dependent penalties (e.g., project delay costs). In this context, the optimizer’s main concern becomes placing each activity at its cost-optimal time, which often means as early as possible to avoid holding or lateness penalties.

If resource constraints (Constraint 17) are still enforced while sequencing is relaxed, then the only factor limiting parallel execution is resource capacity. Activities from the same task or component may overlap freely, as long as the total resource usage does not exceed availability. This transforms the problem into a pure resource allocation problem over time, without any sequencing — similar to multi-processor scheduling with no precedence constraints. Although this version of the problem is still combinatorial due to limited resources, it is far simpler than the original formulation that includes both sequencing and dependency constraints.

By analyzing the optimized Lagrange multipliers $\Gamma=\{\gamma_p\}$, we can determine which activity precedence constraints had the largest impact. High values of γ_p suggest that violating the corresponding constraint (i.e., allowing overlapping or incorrect ordering of activities) yields significant cost savings in the relaxed solution. In practical terms, these may be constraints involving long or resource-intensive activities, where enforcing the correct sequence introduces scheduling delays or additional costs. The dual mechanism compensates for this by penalizing

such violations heavily — effectively signaling that the original constraint was structurally important. The associated Lagrangian dual problem is:

$$\max_{\gamma \geq 0} L(\Gamma) \quad (29)$$

This dual maximization process iteratively adjusts the multipliers γ to strike a balance between the cost reduction from flexible sequencing and the need to approximate the original structure through penalties. However, it's important to note that fully relaxing all sequencing constraints may lead to an overly optimistic lower bound, since it disregards essential procedural requirements. In extreme cases, this relaxation causes the subproblem to decompose per activity, enabling each activity to be scheduled independently at its cheapest possible start time. While this makes the problem extremely fast to solve, the resulting bound may be very loose and not reflective of real-world feasibility.

In our experimental section, we will assess how much tightening or degradation in lower bound quality occurs as a result of relaxing sequencing constraints in isolation. This will help determine whether the computational benefit justifies the potential loss in accuracy.

4.4.2 Relaxing Component Dependency & Resource Capacity Constraints

In addition to relaxing individual sets of constraints, we now explore the combined relaxation of both the resource capacity constraints (Constraint 17) and the component dependency constraints (Constraint 15). The motivation behind this approach is that relaxing multiple constraint sets can significantly simplify the problem structure, potentially enabling complete decomposition and faster lower-bound computation. Constraints (17) and (15) are selected for this purpose because they respectively couple components and time/resource usage — removing both simultaneously decouples the problem in both dimensions.

To implement this relaxation, we will use the two sets of Lagrange multipliers already introduced:

- $\Lambda = \{\lambda_{l,t} \geq 0\}$: for each resource l and time period t , as described in Section 4.4.1
- $M = \{\mu_{ijrk} \geq 0\}$: for each dependency constraint between component I and j , as introduced in Section 4.4.2

The resulting Lagrangian objective function is:

$$L(\Lambda, M) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{t=1}^T \sum_{l=1}^L \lambda_{l,t} \left(\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq (t-D_{jr,k})}^T RN_{jrkl} \times A_{jrk w} - AR_l \right) + \sum_{i,j \in \langle i,j \rangle} \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \mu_{ijrk} \left(\sum_{t=1}^T (t + D_{ir,k}) \times A_{irkt} - \sum_{q=1}^R \sum_{t=1}^T S_{jq t} \times t \right) \right\} \quad (30)$$

In this formulation:

- The resource constraints are dualized via the $\lambda_{l,t}$ multipliers, penalizing any overuse of resource l at time t
- The component dependency constraints are dualized via μ_{ijrk} , penalizing violations where a successor component's task starts before its prerequisite activity completes

The associated dual problem becomes:

$$\max_{\Lambda \geq 0, M \geq 0} L(\Lambda, M) \quad (31)$$

When both component dependencies (Constraint 15) and resource constraints (Constraint 17) are relaxed, the problem becomes substantially simplified — but not fully decomposable — due to the continued enforcement of task-level timing dependencies between components (Constraint 17). These remaining inter-component constraints still link the start times of maintenance tasks across components, preventing full decomposition.

However, significant simplification is still achieved. Without resource capacity limits or detailed activity-level dependencies, the scheduling problem for each component is easier to solve, though not entirely independent. Further relaxation — including of Constraint 14 — would be necessary to achieve full decomposition by component.

This leads to a dramatic simplification: instead of solving one large, integrated MILP, the problem breaks into several smaller subproblems. These smaller problems are often solvable to optimality using greedy methods, brute-force enumeration, or dynamic programming. The overall relaxed solution is simply the union of all these independent component-level schedules.

The resulting lower bound from this combined relaxation is expected to be among the weakest (i.e., lowest) across all relaxation scenarios, as it allows for extensive infeasibility in terms of

resource use and inter-component logic. Conceptually, it approximates the idealized scenario where components could be maintained independently and with unlimited resources — except for the remaining task-level timing dependencies. As such, it provides a highly optimistic estimate of the true optimal cost.

Nonetheless, this bound serves as a valuable baseline. If the cost from the combined relaxation is only marginally lower than that of a resource-only relaxation, it suggests that the component dependencies may not be especially constraining. Conversely, a large gap between the two indicates that both constraint sets — resource capacities and component dependencies — play a critical role in shaping the feasible and cost-effective schedule.

Dual Interpretation and Practical Use

Solving the combined relaxation yields optimal multipliers (Λ^*, M^*) . These provide insight into the interplay between resource and dependency constraints. If a particular component dependency and resource are both tight, we may observe large values for both μ_{ijrk} and $\lambda_{l,t}$, signaling strong constraint interaction. These multipliers can be useful not only for bounding purposes but also to guide heuristic metaheuristic algorithms by identifying which activities are more sensitive to constraint violations.

4.4.3 Relaxing Component Dependency & Activity Sequencing

In this section, we explore the combined relaxation of component dependency constraints (Constraint 15) and activity sequencing constraints (Constraint 16). These two constraint sets govern the logical and temporal relationships between maintenance tasks across and within components. By relaxing them together, we remove both inter-component and intra-task ordering, which can substantially reduce the problem complexity.

To perform this combined relaxation, we use the two sets of Lagrange multipliers introduced before:

- $M = \{\mu_{ijrk} \geq 0\}$: for each dependency constraint in Constraint (15) described in section 4.4.2.

- $\Gamma = \{\gamma_{j,r,a,b} \geq 0\}$: for each sequencing constraint in Constraint (16) described in section 4.4.3.

The Lagrangian objective function for the combined relaxation becomes:

$$L(M, \Gamma) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{i,j \in \langle i,j \rangle} \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \mu_{ijrk} \left(\sum_{t=1}^T (t + D_{irk}) \times A_{irkt} - \sum_{q=1}^R \sum_{t=1}^T S_{jqst} \times t \right) + \sum_{j \in \text{CMP}} \sum_{r \in \text{TSK}} \sum_{(a,b) \in \text{ACT}_{jr} \text{ with } \langle a,b \rangle} \gamma_{j,r,a,b} \times \left[\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} - \sum_{t=1}^T A_{jrbs} \times t \right] \right\} \quad (32)$$

This dualizes both inter-component and intra-task temporal logic, allowing the scheduling algorithm to temporarily ignore dependency and sequencing violations while penalizing them based on the associated multipliers. The corresponding dual problem is:

$$\max_{\mu \geq 0, \gamma \geq 0} L(M, \Gamma) \quad (33)$$

Relaxing both constraints (15) and (16) removes nearly all time-based logic from the model. Inter-component dependencies are ignored, and activities within each task are no longer required to follow a prescribed order. Consequently, the problem decomposes significantly by component and by activity. While resource constraints (Constraint 17) are still enforced in this relaxation, their effect is primarily to limit parallelism.

In this setting, each component's task selection and activity scheduling become nearly independent. Without internal sequencing requirements, the subproblem no longer needs to ensure valid activity progression within tasks. Activities may overlap or be executed out of order, so long as they remain within the planning horizon and adhere to resource availability.

This simplification reduces the original problem from a tightly coupled MILP into a relaxed problem that resembles a multi-mode, resource-constrained scheduling model — but with no ordering restrictions. It becomes feasible to solve each component's subproblem rapidly, especially if the number of tasks and activities per component is small.

From a computational perspective, this formulation offers faster solution times and broader decomposition. However, it is expected to yield a weaker lower bound, as it allows significant infeasibilities. The cost returned by this relaxed model reflects an idealized schedule: one in which

every component may operate on its own timeline, without coordination or adherence to logical task flow.

Dual Interpretation and Practical Use

The optimal multipliers μ^* and γ^* offer insight into which precedence and dependency constraints are most structurally critical. High values signal violations that produce large cost reductions in the relaxed space. These can inform heuristic adjustments or priority rules in future schedule refinement phases.

Solving the dual $\max_{\mu \geq 0, \gamma \geq 0} L(M, \Gamma)$ via subgradient optimization allows us to find the best penalty balance between relaxed dependencies and task orderings. The solutions from this dual form a useful reference for evaluating model sensitivity to logic-driven constraints.

4.4.4 Relaxing Resource Capacity & Activity Sequencing

In this section, we examine the combined relaxation of resource capacity constraints (Constraint 17) and activity sequencing constraints (Constraint 16). These two constraints sets significantly contribute to the complexity of the MRO scheduling problem: one governs the valid execution order of activities within a task, and the other enforces resource feasibility across all activities. Relaxing them together eliminates both task-level temporal dependencies and inter-activity resource competition. To model this relaxation, we use the following sets of Lagrange multipliers:

- $\Lambda = \{\lambda_{l,t} \geq 0\}$: for resource constraints in Constraint (17)
- $\Gamma = \{\gamma_{j,r,a,b} \geq 0\}$: for sequencing constraints in Constraint (15)

The resulting Lagrangian objective function is:

$$L(\Gamma, \Lambda) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{j \in \text{CMP}} \sum_{r \in \text{TSK}} \sum_{(a,b) \in \text{ACT}_{jr} \text{ with } a < b} \gamma_{j,r,a,b} \times \left[\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} - \sum_{t=1}^T A_{jr bt} \times t \right] + \sum_{t=1}^T \sum_{l=1}^L \lambda_{l,t} \left(\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq (t-D_{jrk})}^T RN_{jrkl} \times A_{jrk w} - AR_l \right) \right\} \quad (34)$$

Here, sequencing and resource constraints are both removed from the feasible region and replaced with penalty terms in the objective function. The corresponding dual problem is:

$$\max_{\gamma \geq 0, \lambda \geq 0} L(\Gamma, \Lambda) \quad (35)$$

Relaxing both activity sequencing and resource constraints leads to a highly flexible subproblem. Without sequencing requirements, activities within a task may be scheduled in any order or even concurrently. Without resource capacity limits, there is no restriction on how many tasks or activities can be scheduled simultaneously. The result is a problem that is no longer constrained by either temporal order or shared resource availability. This effectively removes the most computationally intensive elements of the model, allowing the optimizer to focus purely on task selection and activity timing — often defaulting to the earliest start times to reduce delay penalties. For example, the subproblem may schedule all activities at the beginning of the planning horizon if there are no other constraints limiting it. This relaxation can produce extremely optimistic lower bounds, since it ignores fundamental real-world limitations. The model may return a schedule that is not only infeasible but physically impossible. However, it offers valuable benchmarking insight: it reveals how much cost is introduced by enforcing just sequencing and resource feasibility.

Dual Interpretation and Practical Use

Solving the Lagrangian dual $\max_{\gamma \geq 0, \lambda \geq 0} L(\Gamma, \Lambda)$ reveals how costly it is to enforce sequencing and resource feasibility. High multipliers $\gamma_{j,r,a,b}$ identify activity orderings that strongly influence cost, while high $\lambda_{l,t}$ values indicate resource bottlenecks. Observing how these evolve during subgradient optimization highlights which constraints are most restrictive in practical terms. From an implementation standpoint, this combined relaxation is useful in early-stage exploratory analyses or as a component in heuristic or metaheuristic algorithms. While the bound may not be tight, it is computed quickly and can be used to rank the impact of constraint sets.

4.4.5 Relaxing Resource Capacity, Component Dependency, & Activity Sequencing

In this final relaxation scenario, we explore the combined relaxation of resource capacity constraints (Constraint 17), component dependency constraints (Constraint 15), and activity sequencing constraints (Constraint 16). Together, these three constraint sets define the core structure of the original MRO scheduling model, capturing inter-component logic, intra-task execution order, and shared resource limitations. Relaxing all three yields the most decomposed

and computationally tractable version of the problem. We use the following sets of Lagrange multipliers:

- $M = \{\mu_{ijrk} \geq 0\}$: for each dependency constraint in Constraint (15)
- $\Gamma = \{\gamma_{j,r,a,b} \geq 0\}$: for sequencing constraints in Constraint (16)
- $\Lambda = \{\lambda_{l,t} \geq 0\}$: for resource constraints in Constraint (17)

The corresponding Lagrangian objective function is:

$$L(M, \Gamma, \Lambda) = \min_x \left\{ \text{OriginalObjective}(x) + \sum_{i,j \in \langle i,j \rangle} \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \mu_{ijrk} \left(\sum_{t=1}^T (t + D_{irk}) \times A_{irkt} - \sum_{q=1}^R \sum_{t=1}^T S_{jq} \times t \right) + \sum_{j \in \text{CMP}} \sum_{r \in \text{TSK}} \sum_{(a,b) \in \text{ACT}_{jr} \text{ with } \langle a,b \rangle \gamma_{j,r,a,b}} \gamma_{j,r,a,b} \times \left[\sum_{t=1}^T (t + D_{irk}) \times A_{jrat} - \sum_{t=1}^T A_{jrbt} \times t \right] + \sum_{t=1}^T \sum_{l=1}^L \lambda_{l,t} \left(\sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq (t - D_{jrk})}^T RN_{jrkl} \times A_{jrk w} - AR_l \right) \right\} \quad (36)$$

The associated dual problem is:

$$\max_{\mu \geq 0, \gamma \geq 0, \lambda \geq 0} L(M, \Gamma, \Lambda) \quad (37)$$

Relaxing all three constraint types results in a highly decoupled scheduling problem. Without logical dependencies, task sequencing, or resource competition, much of the original structural coupling is removed. Each component and activity become significantly easier to optimize, often independently or in small groups. However, complete decoupling is not achieved, as Constraint (14), which enforces start-time alignment across components, remains in effect. This constraint still requires some level of coordination between components when determining when their maintenance tasks begin, particularly when such timing relationships are necessary (e.g., for fleet-level planning or interdependent component readiness). As a result, while the problem becomes far more tractable, it cannot be solved purely through single-activity or fully isolated component-level optimizations. In this setting, activities may be scheduled out of order and without restriction on resource usage or task dependencies. The model effectively becomes a cost-driven activity placement exercise, constrained only by start-time alignment rules. This simplification enables much faster computation and tighter decomposition than in the full model but still requires consideration of cross-component timing. However, this extreme relaxation also yields the loosest lower bound of all relaxation scenarios. The solution no longer reflects any real-world feasibility and instead provides a theoretical minimum cost — one that would be achievable only in an ideal world with no physical limitations or coordination requirements.

Despite this, the bound retains strategic value. It provides a baseline for comparison and helps isolate the aggregate impact of the model’s structural constraints. If the cost reduction from this relaxation is only marginal compared to partial relaxations, it may indicate that some constraints are weakly binding. In contrast, a large gap would suggest strong interdependence between components, activities, and resources in the original formulation.

Dual Interpretation and Practical Use

Solving the combined dual $\max_{\mu \geq 0, \gamma \geq 0, \lambda \geq 0} L(M, \Gamma, \Lambda)$ produces a full set of multipliers that encode the "pressure" each constraint set exerts on the schedule. This helps identify which constraints are most costly to violate and guides prioritization in tightening or heuristic repair. Although the resulting schedule is infeasible by design, the solution from this relaxation can serve as a foundation for metaheuristic construction or feasibility repair heuristics. Additionally, this scenario provides the fastest possible bound for performance benchmarking.

From an algorithmic standpoint, the dual problems resulting from these relaxations are solved using the subgradient optimization method, which is detailed in the following section. Beyond its role in computing lower bounds, the structure of the relaxed subproblems—particularly those that decompose by component—also provides a foundation for heuristic solution construction. While our primary approach for generating high-quality feasible solutions remains the Iterative Local Search method, the subgradient-based duals offer valuable insights and initialization strategies for such heuristics.

4.5 Subgradient Optimization Framework

Solving the Lagrangian dual – that is, finding the optimal Lagrange multipliers for the relaxed constraints – is a challenging task due to the nondifferentiability of the dual function. We employ a subgradient optimization method to iteratively update the multipliers and maximize the Lagrangian bound. Subgradient methods are well-suited for such nonsmooth optimization problems, especially when dealing with the piecewise-linear concave duals resulting from the Lagrangian relaxation of integer programming models.

This section describes the subgradient optimization framework used throughout all relaxation scenarios in this study, covering the full set of constraints we relaxed: resource capacity (Constraint 17), component dependency (Constraint 15), and activity sequencing (Constraint 16).

4.5.1 Initialization

The subgradient method begins with initializing all Lagrange multipliers to zero:

$$\lambda_{l,t}^{(0)} = 0, \mu_{i,j,r,k}^{(0)} = 0, \gamma_{j,r,a,b}^{(0)} = 0 \quad (38)$$

Where $\lambda_{l,t}$ is defined for all resources $l \in RES$ and time periods $t \in TME$, $\mu_{i,j,r,k}$ for each dependency constraint between components i and j , and $\gamma_{j,r,a,b}$ for sequencing constraints within tasks for component j .

The initialization of multipliers affects the trajectory of the optimization path. While subgradient methods are generally robust to the starting point, poor initialization may result in slow convergence or instability during early iterations.

In some tests, we experimented with heuristic-based initialization — for example, setting $\lambda_{l,t}^{(0)}$ proportional to the resource overload observed in an initial feasible solution produced by the ILS heuristic. Although such strategies sometimes improved early progress, they did not significantly impact the final bounds. Therefore, for consistency and reproducibility, we used zero initialization in all reported results unless otherwise noted. At each iteration q , the method proceeds as follows:

4.5.2 Solving the Relaxed Subproblems

We solve the Lagrangian-relaxed version of the original problem using the current set of multipliers $(\Lambda^{(q)}, M^{(q)}, \Gamma^{(q)})$. All subproblems, regardless of which constraints are relaxed, are formulated as MILP models in Python and solved using the CPLEX solver. The degree of decomposition varies depending on the combination of constraints relaxed:

- If only resource constraints (Constraint 17) are relaxed, the subproblem remains coupled through dependencies and sequencing.

- If only dependency constraints (Constraint 15) are relaxed, the problem partially decomposes by component and each subproblem is solved exactly via CPLEX.
- If only sequencing constraints (Constraint 16) are relaxed, the problem is a resource-constrained scheduling problem without sequencing, again solved via MILP in CPLEX.
- For combinations, the subproblem may partially or fully decompose, and is solved as a reduced MILP.

4.5.3 Subgradient Calculation

For each relaxed constraint, we compute the subgradient component based on the violation of the constraint in the relaxed solution $x^{(q)}$:

- Resource constraint:
$$v_{l,t}^{(q)} = \sum_{j=1}^J \sum_{r=1}^R \sum_{k=1}^{K_{jr}} \sum_{w \geq (t-D_{jrk})}^T RN_{jrkwl} \times A_{jrkwl}^{(q)} - AR_l \quad (39)$$

- Component dependency constraint:
$$d_{i,j,r,k}^{(q)} = \sum_{t=1}^T (t + D_{irk}) \times A_{irklt} - \sum_{q=1}^R \sum_{t=1}^T S_{jqt}^{(q)} \times t \quad (40)$$

- Activity sequencing constraint:
$$s_{j,r,a,b}^{(q)} = \sum_{t=1}^T (t + D_{irk}) \times A_{jrat} - \sum_{t=1}^T A_{jrbt}^{(q)} \times t \quad (41)$$

The subgradient vector $g^{(q)}$ is composed of all constraint violation values. Positive subgradient values indicate constraint violations to be penalized.

4.5.4 Subgradient Norm Computation

We use the Euclidean (L2) norm to measure the magnitude of the subgradient:

$$\|g^{(q)}\| = \sqrt{\sum_{l,t} (v_{l,t}^{(q)})^2 + \sum_{i,j,r,k} (d_{i,j,r,k}^{(q)})^2 + \sum_{j,r,a,b} (s_{j,r,a,b}^{(q)})^2} \quad (42)$$

This norm controls the magnitude of the multiplier updates in the next step.

4.5.5 Multiplier Update

We update the multipliers based on the subgradient direction:

$$\lambda_{l,t}^{(q+1)} = \max\{0, \lambda_{l,t}^{(q)} + \alpha^{(q)} v_{l,t}^{(q)}\} \quad (43)$$

$$\mu_{ijrk}^{(q+1)} = \max\{0, \mu_{ijrk}^{(q)} + \alpha^{(q)} d_{ijrk}^{(q)}\} \quad (44)$$

$$\gamma_{jrab}^{(q+1)} = \max\{0, \gamma_{jrab}^{(q)} + \alpha^{(q)} s_{jrab}^{(q)}\} \quad (45)$$

The max operator ensures that all multipliers remain non-negative, consistent with the fact that the relaxed constraints are of the “ $\leq$ ” type.

4.5.6 Step Size Selection

We use the Polyak step size rule:

$$\alpha^{(q)} = \epsilon \cdot \frac{(UB-LB^{(q)})}{\|g^{(q)}\|^2} \quad (46)$$

Where:

- UB : Upper bound from a feasible solution (ILS heuristic)
- $LB^{(q)}$: Lagrangian bound from iteration q
- ϵ : A scalar in $(0,2]$; adjusted to control convergence

This formulation ensures larger steps when far from the optimum, and smaller steps as the duality gap closes.

4.5.7 Convergence Criteria

We stop the iterations if one of the following holds:

- Maximum number of iterations is reached (200 iteration in our study)
- The gap falls under a specific threshold (set to 1% in our study)

All Lagrangian subproblems are implemented in Python and solved using IBM CPLEX. Multiplier evolution is monitored, with large values of indicating scheduling bottlenecks. We also monitor the evolution of the Lagrange multipliers during the optimization process. In particular, large values of resource multipliers $\lambda_{l,t}$ often indicate resource bottlenecks in the schedule, while high values of $\mu_{i,j,r,k}$ or $\gamma_{j,r,a,b}$ reveal particularly tight dependency or sequencing constraints. While

exact convergence is not guaranteed due to the non-differentiability of the dual function, the method reliably produces high-quality bounds that allow effective assessment of the impact of each relaxation scenario.

4.5.8 Implementation Details

We coded the subgradient method in Python, interfacing with a MILP solver (CPLEX) for solving all Lagrangian subproblems, regardless of the problem size or the degree of decomposition. One important detail is feasibility restoration: at the end of the subgradient procedure, we have a set of multipliers and a corresponding relaxed solution that may violate constraints. While the multipliers give a bound, we also want a feasible schedule solution. We relied on the ILS heuristic to produce a feasible solution and compute the initial upper bound (UB) prior to running subgradient. Since we do not update UB during subgradient iterations, the method is used solely for lower bound computation and analysis. While one could use primal heuristics on the dual — e.g., taking the schedule from the Lagrangian solution and adjusting it to fix violations — we focused on leveraging ILS for feasible scheduling and subgradient solely for bounding purposes.

In summary, the subgradient framework allowed us to solve the Lagrangian duals for each relaxation scenario: we obtained a sequence of improving lower bounds $LB^{(0)} \leq LB^{(1)} \leq \dots$ that converge toward a maximum, and the final multipliers that achieve those bounds. With these in hand, we proceed to analyze the numerical results, comparing the different relaxations and assessing the quality of the bounds relative to heuristic solutions.

4.6 Numerical Results and Discussion

We conducted computational experiments to evaluate the performance of the Lagrangian relaxation and subgradient optimization approach on the MRO scheduling problem. The experiments were designed to address several key questions:

1. How tight are the lower bounds produced by relaxing different sets of constraints when compared to the best-known feasible solutions (upper bounds)?
2. How quickly does the subgradient method converge for each relaxation scenario, and what are the characteristics of the convergence behavior?

3. For small problem instances where the true optimal solution can be obtained by solving the original MILP to optimality using CPLEX, how close are the Lagrangian bounds to the optimal objective, and do they correctly identify the optimum?
4. For larger instances, where exact optimal solutions are not tractable, what do the Lagrangian bounds reveal about the quality of the feasible solutions produced by the ILS heuristic, and about the relative impact of various constraint sets on the total objective?

In all cases, the computed Lagrangian lower bounds are compared to a fixed upper bound UB, derived either from the exact optimal MILP solution (for small instances) or from the best ILS solution (for large instances), as described in the previous section. This ensures consistency in the interpretation of duality gaps across all relaxation configurations.

4.6.1 Test Instances

We tested our Lagrangian relaxation and subgradient optimization approach on a range of problem instances, from small-scale scenarios to realistically large cases.

- Small instances consisted of 7 components and a short planning horizon. These instances were solved to optimality using the original MILP model and CPLEX solver, providing ground-truth optimal objective values for benchmarking the Lagrangian lower bounds. We used the same data set introduced in first chapter.
- Large instances involved 32 components, multiple resource types, and a long planning horizon that resulted in hundreds of activities. For these large instances, exact optimization was computationally infeasible. Therefore, we used the best feasible solutions from the ILS heuristic as the upper bound and applied the Lagrangian relaxation approach to compute corresponding lower bounds. The same data set used in chapter 2 is used here as well.

All instances included two types of resources: machines and labor. Additionally, each instance included two dummy components—a dummy start and a dummy finish node—used as modeling constructs to ensure proper sequencing and logical structure in the activity network. These dummy components do not represent physical parts of the aircraft but serve to simplify constraint modeling.

4.6.2 Relaxation Schemes

We applied a total of seven Lagrangian relaxation schemes, corresponding to different combinations of constraint relaxations, as outlined in Section 4.4:

- A. LR-Res: Relax resource constraints only (Constraint 17)
- B. LR-Dep: Relax component dependency constraints only (Constraint 15)
- C. LR-Seq: Relax activity sequencing constraints only (Constraint 16)
- D. LR-Dep+Res: Relax both resource and component dependency (Constraints 17 + 15)
- E. LR-Dep+Seq: Relax both component dependency and activity sequencing constraints (Constraints 15 + 16)
- F. LR-Seq+Res: Relax both resource and activity sequencing constraints (Constraints 17 + 16)
- G. LR-Dep+Seq+Res: Relax all three constraints (Constraints 17 + 15 + 16)

For each of these relaxation schemes, we applied the subgradient optimization algorithm to compute a corresponding lower bound on the optimal objective value. All subgradient runs were conducted for 200 iterations, with the step size parameter (ϵ) set to 0.7 in all cases.

In the results that follow, we first present the outcomes for the small instances (where true optimal solutions are known), followed by the results for the large instances (where bounds are compared against the best feasible ILS solutions).

4.6.3 Results on Small Instances

To evaluate the performance of the proposed Lagrangian Relaxation (LR) method and better understand the effects of relaxing different combinations of constraints on solution quality and convergence behavior, a series of numerical experiments were conducted on a small instance of the maintenance scheduling problem. This controlled example enables a detailed assessment of the LR approach within a manageable setting where the optimal solution is known.

For the small instance, the optimal solution obtained via CPLEX is \$166,262.75. This instance corresponds to the same dataset introduced in Chapters 2 and 3. In all experiments, the subgradient method was allowed to run without a time limit — it proceeded until the maximum number of iterations was reached, uniformly across all relaxation schemes.

At each iteration, the objective value and relative gap to the known optimal solution were tracked using the following formula:

$$Gap = \frac{|Z_{Current} - Z_{Optimal}|}{Z_{Optimal}} \times 100 \quad (47)$$

Although the Lagrangian relaxation yields lower bounds (and therefore raw logs may display negative gaps), we report the absolute value to simplify the comparison across scenarios — smaller gap values indicate better solution quality.

Relaxation Scenarios

Seven schemes were tested, each corresponding to a specific combination of relaxed constraints, as shown in Table 4.1.

Table 4.1 Relaxation schemes for subgradient optimization

Scheme	Relaxed Constraint	Notes
A	LR-Res	Resource constraints (Constraint 17)
B	LR-Dep	Component dependencies (Constraint 15)
C	LR-Seq	Activity sequencing (Constraint 16)
D	LR-Dep+Res	Combined relaxation of 17 & 15
E	LR-Dep+Seq	Combined relaxation of 15 & 16
F	LR-Seq+Res	Combined relaxation of 17 & 16
G	LR-Dep+Seq+Res	Combined relaxation of 17, 15 & 16

Table 4.2 below summarizes the key results for all scenarios, including best gap achieved and total computational time:

Table 4.2 Summary of results for each scheme

Scheme	Best Gap (%)	Total Time (seconds)
A	7.94	2239.6
B	1.17	14542.1
C	20.16	4174.1
D	23.46	817.2
E	26.16	1941
F	14.47	4162.4
G	31.11	587.2

Comparative Convergence Analysis

Several key insights emerge from analyzing the convergence behavior and final gap values:

1. Impact of Relaxing Individual Constraints:

- Scenario B (Constraint 15 only): Yields the tightest gap (1.17%) but incurs the longest runtime (>14,500 seconds), highlighting the dominant role of component dependency constraints in determining schedule feasibility.
- Scenario A (Constraint 17 only): Achieves a good balance with a 7.94% gap and moderate runtime (~2,240 seconds), showing resource constraints are influential but more manageable.
- Scenario C (Constraint 16 only): Produces a looser gap (20.16%) in ~4,174 seconds, indicating sequencing contributes less critically to solution quality.

2. Impact of Combining Relaxations:

- Scenario G (All three constraints): Yields the fastest convergence (~587 seconds) but also the worst gap (31.11%), consistent with maximal infeasibility due to full decoupling.
- Scenario D and E (15 + 17, 15 + 16): Yield moderate-to-high gaps (23.46%, 26.16%) with significant time savings over relaxing Constraint 15 alone.
- Scenario F (17 + 16): Provides a strong trade-off — 14.47% gap and ~4,162 seconds — suggesting that relaxing 17 and 16 together balances quality and speed effectively.

3. General Trends:

- Relaxing more constraints reduces runtime but typically worsens the bound.
- Relaxing Constraint 15 (dependencies) alone gives the tightest gap but is computationally expensive.
- Constraint 16 (sequencing) appears to have less impact on bound quality but helps reduce runtime.

Gap Profile Analysis Across Scenarios

To further understand the convergence dynamics of the proposed Lagrangian Relaxation (LR) method across different relaxation scenarios, this section presents a detailed examination of the

gap evolution over the iterations. The gap, representing the relative distance between the current solution and the optimal lower bound, is a key indicator of both convergence speed and stability of the LR process.

Figure 4.1 through Figure 4.7 illustrate the gap progression plots for all seven scenarios corresponding to the small instance.

Scenario A (Relaxing Constraint 17)

This scenario demonstrates the most stable convergence among all tested relaxations. After an initial sharp drop in the gap during the first ~10 iterations (from approximately 14% to around 9%), the algorithm consistently refines the solution, with the gap gradually decreasing and stabilizing between 8% and 9%. Importantly, while some minor oscillations persist, no large fluctuations or spikes occur throughout the subsequent 190 iterations, indicating a robust and well-conditioned optimization trajectory. This behaviour suggests that Constraint 17 is relatively well-suited to be treated within the Lagrangian relaxation framework, leading to a controlled and predictable convergence pattern.

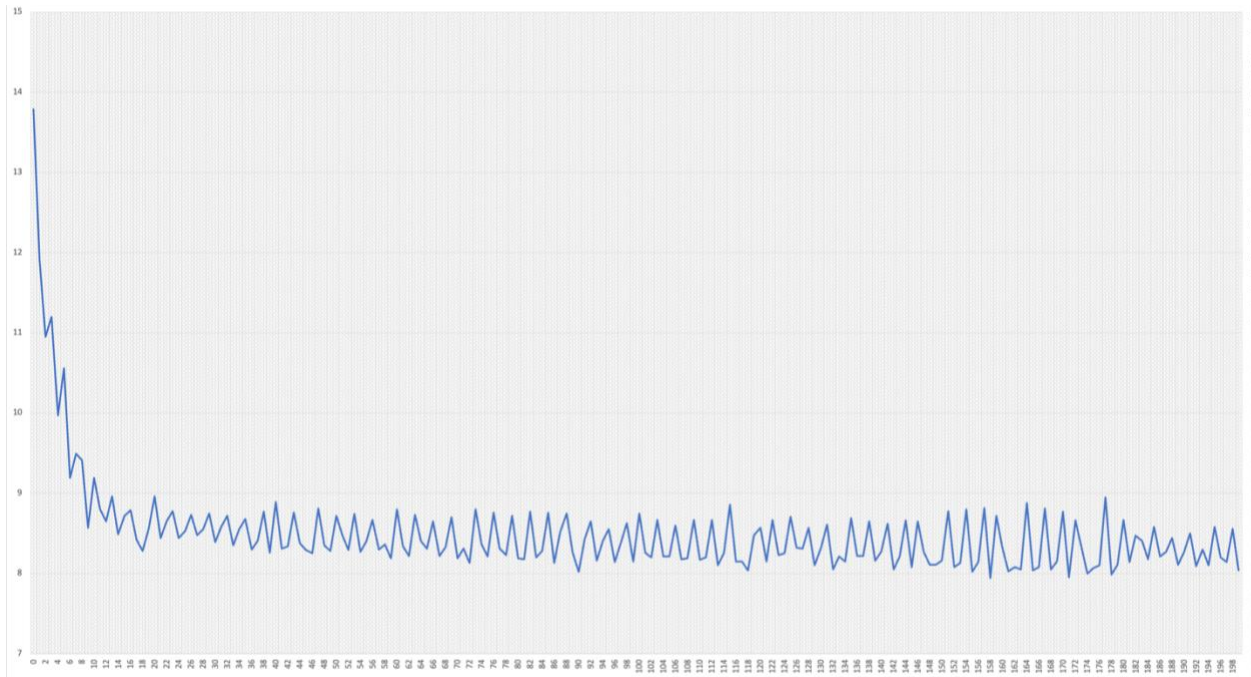


Figure 4.1 Gap Progression Plot – Scenario A

Scenario B (Relaxing Constraint 15)

This scenario exhibits a highly stable convergence profile, with the gap oscillating tightly within a narrow band between approximately 1.15% and 1.4%. While no clear long-term trend of reduction is observed, the stability of the gap across all iterations reflects a well-balanced multiplier update process. The near-flat behaviour indicates that Constraint 15, which governs intra-component activity precedence, is not excessively disruptive to the underlying problem structure when relaxed, resulting in a smooth and stable optimization process. Furthermore, the extremely low final gap in this scenario (1.17%) is the best among all scenarios, underscoring both the tractability of this relaxation and the strength of the solution quality.

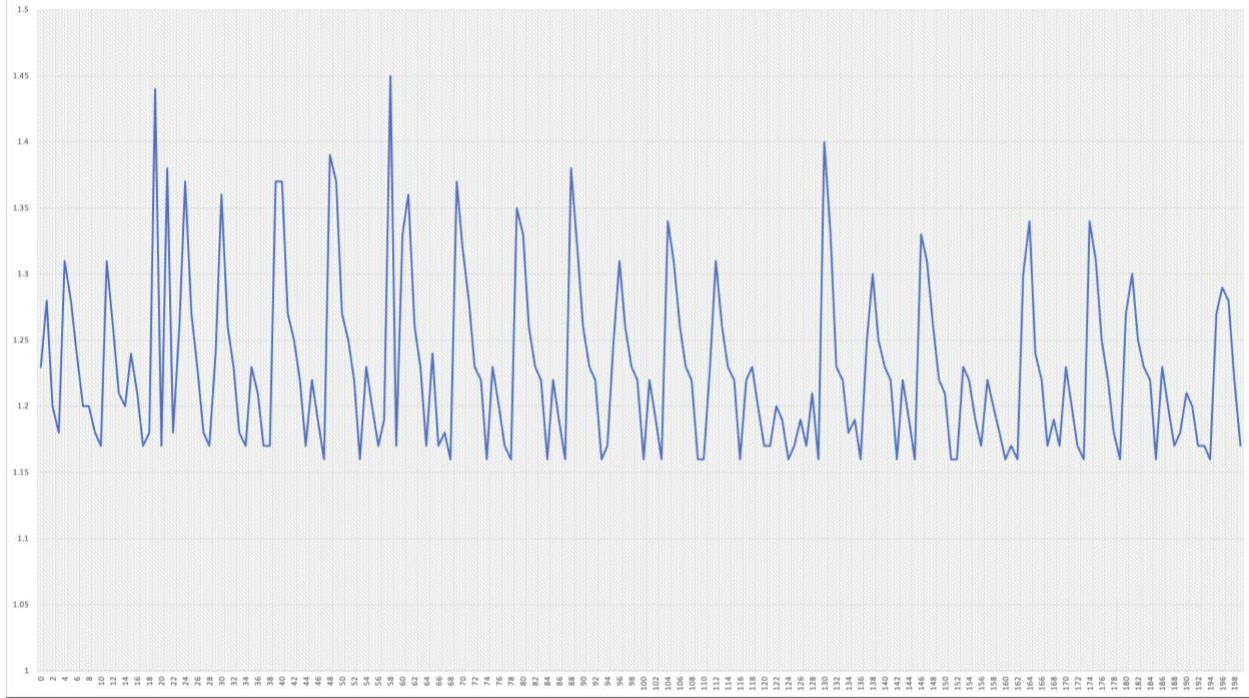


Figure 4.2 Gap Progression Plot – Scenario B

Scenario 3 (Relaxing Constraint 16)

In contrast to the previous two cases, Scenario C presents a more challenging convergence behaviour. While an initial reduction in the gap is observed during the early iterations (from approximately 62% to around 30%), the algorithm fails to fully stabilize thereafter. The gap exhibits strong periodic oscillations, with repeated cycles ranging from ~22% to ~40%, and occasional larger spikes. These persistent fluctuations suggest a more complex interaction between Constraint 26 and the rest of the problem structure. The dynamics imply that relaxing these

precedence relationships introduces more instability into the dual problem, likely due to the strong coupling between activities. As a result, convergence is slower and less stable, and the final best gap remains at a relatively high value of 20.16%.

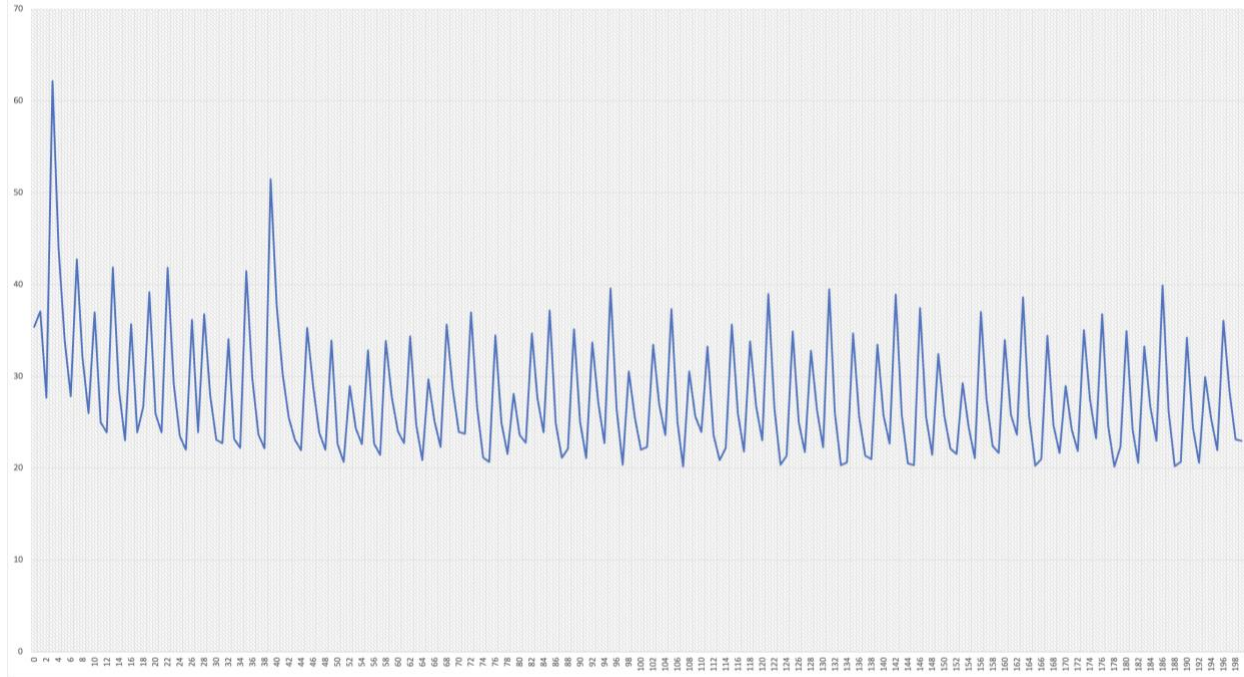


Figure 4.3 Gap Progression Plot – Scenario C

Scenario D (Relaxing Constraints 15 and 16)

This scenario exacerbates the instability observed in Scenario C. The combined relaxation of task dependency and intra-component precedence constraints leads to highly irregular behaviour throughout the iterations. The gap fluctuates across a wide range (~40% to ~80%) with no discernible downward trend, and several large spikes are present even in later iterations. The apparent worsening of stability when adding Constraint 16 to the relaxation of Constraint 15 suggests that the simultaneous relaxation of two structurally important precedence-based constraints may overload the dual optimization process, reducing the ability of the algorithm to control and correct the multipliers effectively. This is reflected in the final gap of 23.46%, with no clear signs of convergence plateau.

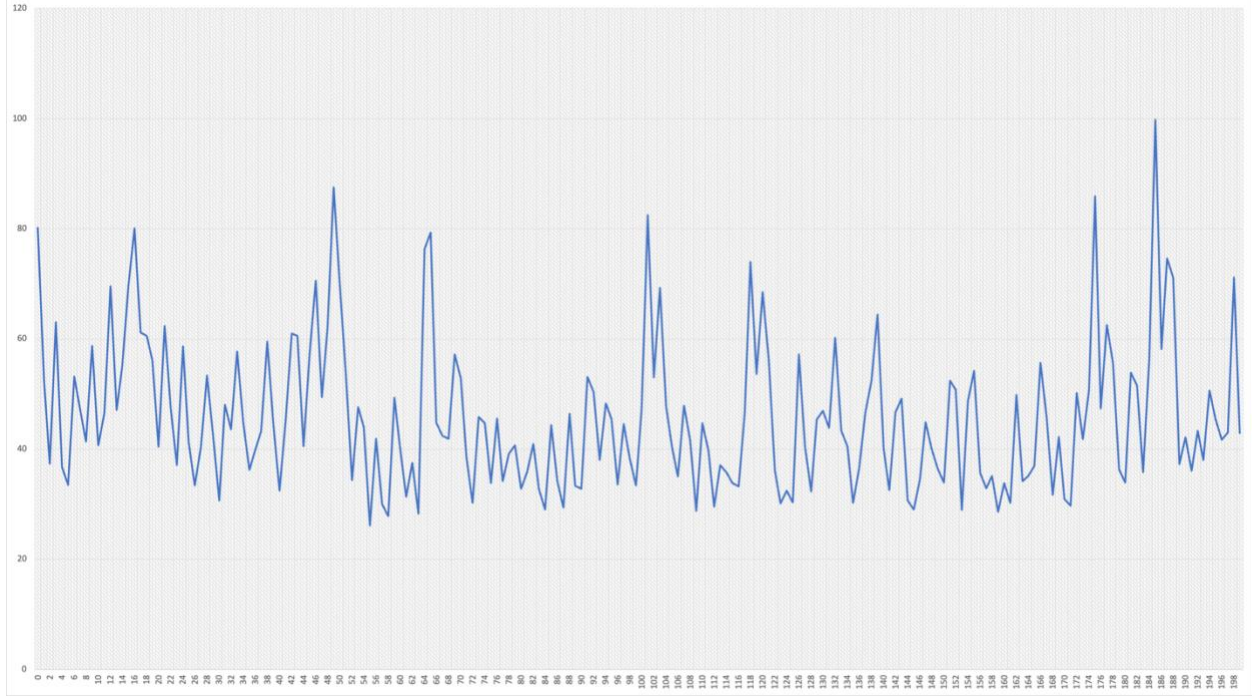


Figure 4.4 Gap Progression Plot – Scenario D

Scenario E (Relaxing Constraints 15 and 17)

The relaxation of both task dependencies and resource constraints leads to highly volatile convergence dynamics. The gap oscillates severely across the entire range of iterations, with frequent sharp spikes exceeding 100% and valleys dropping to around 20%-30%. Moreover, the frequency of large spikes increases in certain segments of the iterations (notably around iterations 30–80 and 100–160), indicating instability in the underlying multiplier updates. This behaviour is indicative of conflicting interactions between the relaxed resource constraints and task sequencing dependencies, both of which are highly influential in shaping feasible solutions. The final best gap of 26.16% reflects this difficulty, making this one of the least effective relaxations tested.

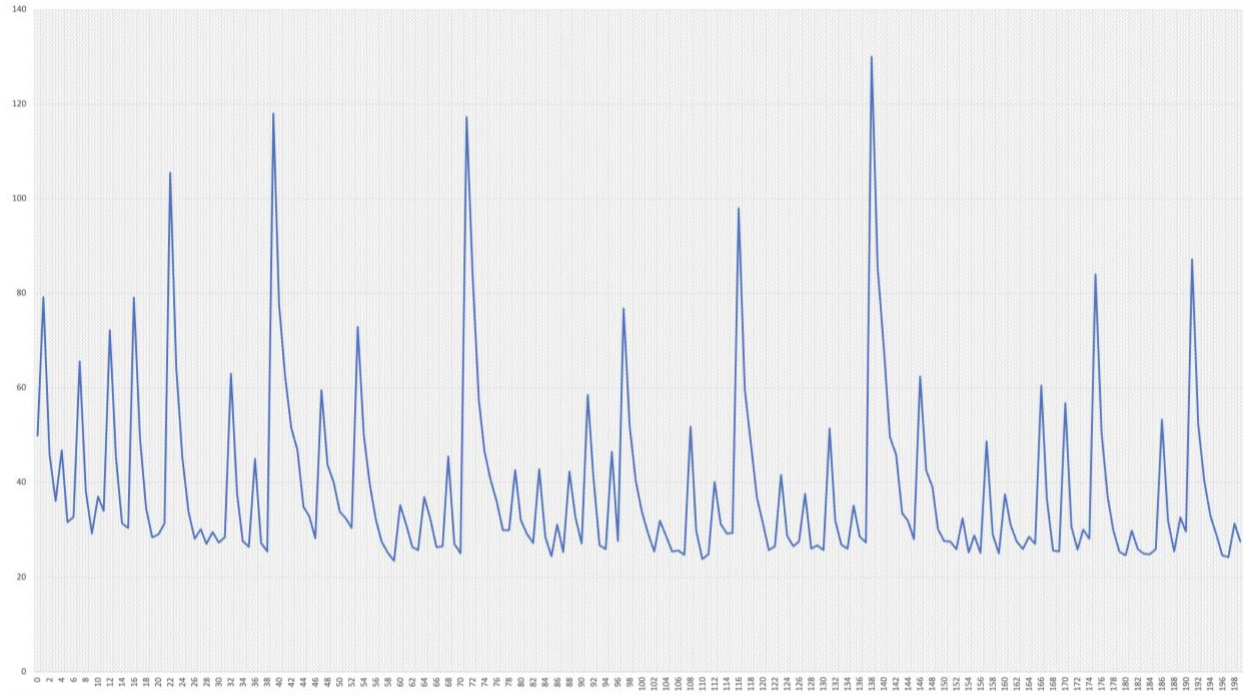


Figure 4.5 Gap Progression Plot – Scenario E

Scenario F (Relaxing Constraints 16 and 15)

This scenario also exhibits considerable instability, with the gap showing strong oscillations and repeated large peaks, even in late iterations. The gap frequently exceeds 70% and oscillates down to ~20% or lower, with no stable convergence phase emerging. Although both individual relaxations (Scenario A and Scenario B) were relatively stable, their combination here results in significant interaction effects, disrupting the multiplier optimization. This suggests that the precedence constraints (Constraint 16) and resource constraints (Constraint 17) may become more tightly coupled when relaxed simultaneously, undermining convergence control. The final best gap of 14.47% is respectable but achieved through an unstable and inefficient trajectory.

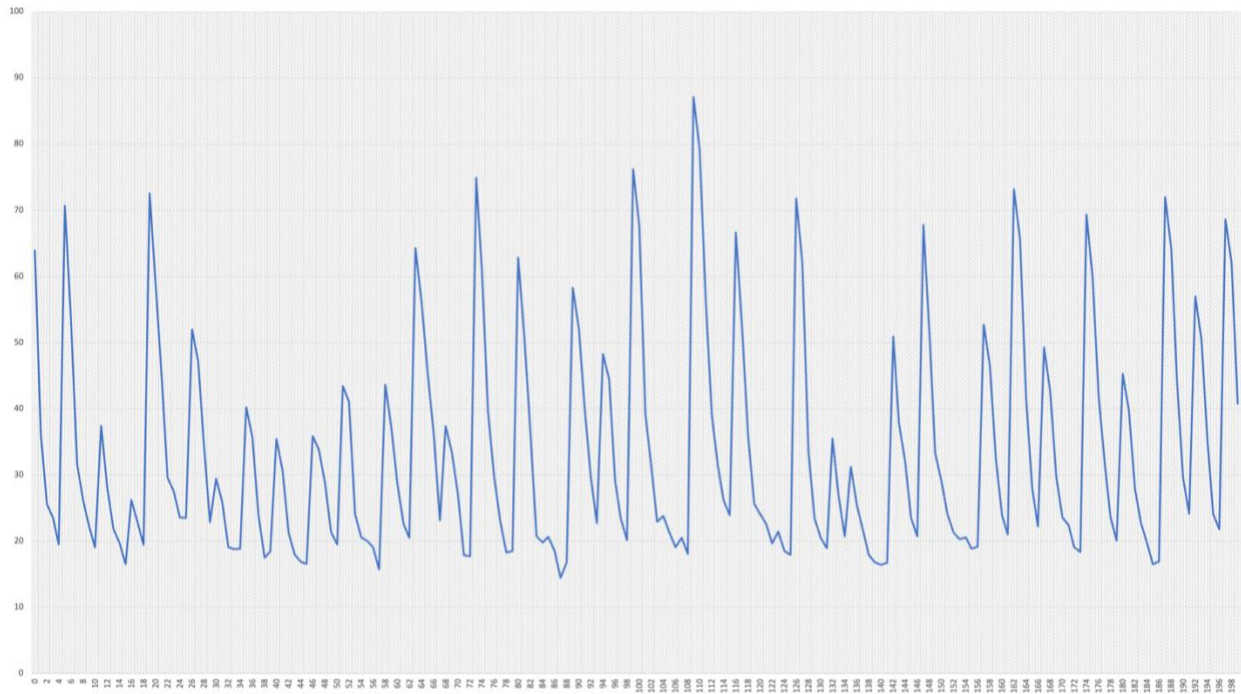


Figure 4.6 Gap Progression Plot – Scenario F

Scenario G (Relaxing Constraints 17, 15, and 16)

Scenario G represents the most unstable case among all scenarios. The gap starts at an extremely high level (~370%) and undergoes massive oscillations during the first ~30 iterations. Although it eventually stabilizes into a lower regime (~50%-100%), large spikes continue to occur intermittently throughout the entire run, including in the final 50 iterations. The combination of all three key structural constraints leads to severe interaction effects, as precedence, sequencing, and resource allocation become entirely unconstrained in the primal model, leaving the Lagrangian process highly sensitive to multiplier updates. Despite the fastest total solution time (587.2 seconds), this scenario yields the worst final best gap (31.11%), highlighting a classic trade-off between speed and solution quality when over-relaxing critical constraints.

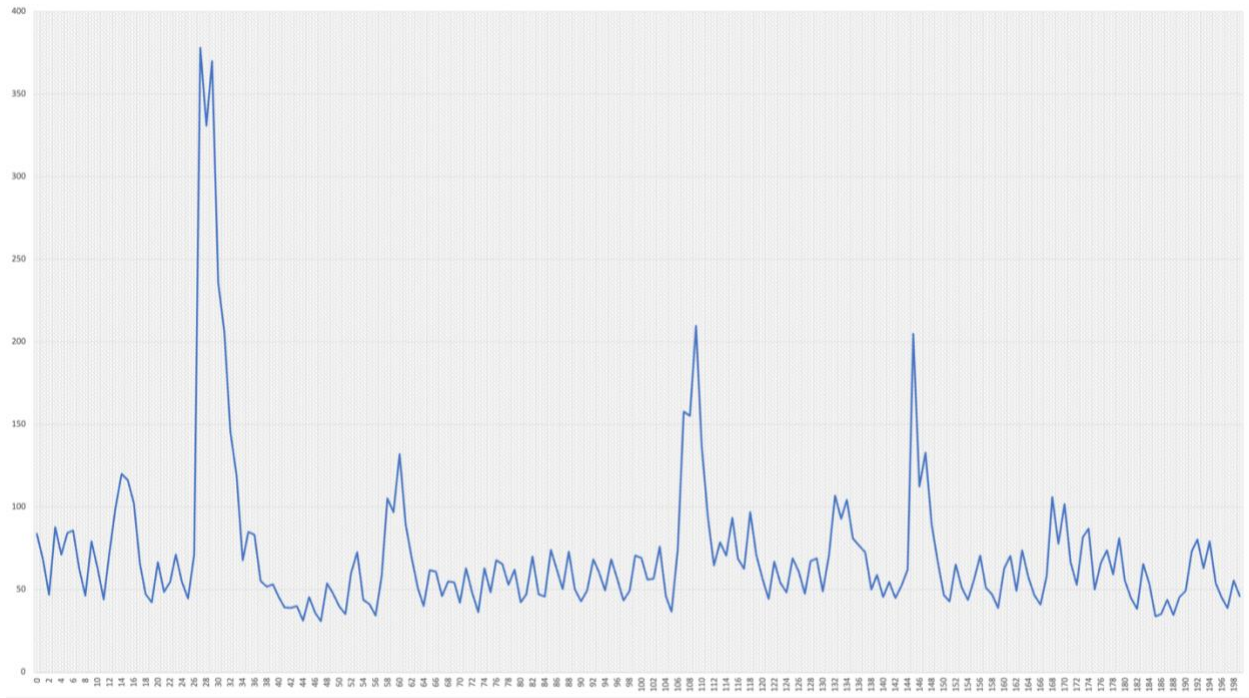


Figure 4.7 Gap Progression Plot – Scenario G

4.6.4 Overall Interpretation and Conclusions

The analysis of gap evolution across all scenarios reveals important insights into the behaviour of the Lagrangian relaxation method when applied to this class of maintenance scheduling problems.

First, relaxing only Constraint 17 or Constraint 16 leads to stable and effective convergence, producing both strong gap performance and controlled optimization dynamics. These constraints appear to interact relatively benignly with the core problem structure when relaxed.

In contrast, relaxing Constraint 15—either alone or in combination—introduces significant instability, with large oscillations and slower convergence. This highlights the critical role of task dependencies in maintaining structural coherence in the schedule and suggests that this constraint is not well-suited to pure Lagrangian relaxation without additional stabilization mechanisms (e.g., adaptive epsilon, hybrid decomposition, or heuristic guidance).

The combination of multiple relaxed constraints generally worsens both stability and gap performance, as seen in Scenarios D, E, D, and G. Notably, the most extreme instability is observed when all three constraints are relaxed, confirming that over-relaxation can overload the dual optimization process.

In terms of practical implications, this analysis suggests that careful selection of relaxed constraints is essential for balancing convergence stability, gap quality, and computational efficiency. For this problem class, relaxing only Constraint 16 or Constraint 17 offers a favourable trade-off, while combinations involving Constraint 25 require further refinement to achieve acceptable stability. Table 4.2 presents the best gap achieved in each scenario and the corresponding runtime for 200 iterations.

4.6.5 Gap Evolution and Stability – Adaptive Epsilon

Following the gap behaviour analysis presented in the previous section, the current section explores approaches to improve the stability of the gap evolution within the Lagrangian relaxation framework. Rather than applying these stabilization techniques across all scenarios, a representative scenario is selected to provide a focused and controlled evaluation of their effectiveness.

Scenario D, corresponding to the relaxation of Constraints 15 and 17, is chosen for this purpose. This scenario demonstrates several favourable characteristics that make it an ideal candidate for stability testing. First, among the scenarios involving combined relaxations, Scenario D achieves an attractive trade-off between solution quality and computational efficiency: it yields a best gap of 23.46%, while requiring only 817.2 *seconds* of computational time — the shortest runtime among all multi-constraint relaxations tested. Second, as illustrated in the previous gap evolution analysis (Figure 4.4), Scenario D also exhibits high-frequency fluctuations and pronounced spikes in the gap trajectory, indicating a significant opportunity for improvement. The presence of such oscillations makes this scenario particularly suitable for testing stabilization strategies aimed at smoothing convergence.

One of the primary factors contributing to the observed instability is the use of a constant epsilon (ϵ) value in the subgradient optimization process. While a fixed epsilon is simple to implement, it does not account for the dynamic nature of the optimization trajectory. In the early stages of the algorithm, larger steps may be beneficial for rapid progress, while in later stages, smaller, more refined updates are often required to achieve precise convergence. Without such adaptation, the fixed epsilon may induce oscillations and prevent the method from reaching a stable and optimal solution trajectory. To address this, an adaptive epsilon adjustment mechanism is introduced,

aimed at promoting both convergence stability and solution quality. Specifically, the epsilon update rule is modified as follows:

$$\epsilon_k = \frac{\epsilon_0}{1+\alpha k} \quad (48)$$

where:

- ϵ_0 is the initial epsilon value, set to the same constant used in the baseline runs,
- k is the current iteration number,
- α is a small positive parameter controlling the rate of decay.

This formulation introduces a progressive decay in epsilon across iterations, enabling the algorithm to take larger steps initially (to escape poor local regions and promote global exploration), followed by gradually smaller, more conservative steps as convergence is approached. This adaptive epsilon-decay strategy is consistent with approaches in the literature, such as [\(Pang et al., 2017\)](#), who introduced an explicit epsilon-decay mechanism in subgradient optimization to stabilize convergence in Lagrangian relaxation of complex industrial scheduling problems

The impact of this adaptive epsilon approach on Scenario D will be systematically evaluated in the next section. Specifically, a direct comparison will be made between the original fixed-epsilon results and the new adaptive-epsilon results, focusing on three key aspects:

1. Gap evolution stability — reduction of oscillations and spikes.
2. Final solution quality — improvement or maintenance of best gap achieved.
3. Convergence speed — changes in runtime and convergence profile.

The findings of this experiment will offer insights into the practical benefits and applicability of adaptive strategies within the broader Lagrangian framework developed in this study.

Adaptive Epsilon Results

The performance of the adaptive epsilon strategy was evaluated on Scenario D (relaxing Constraints 15 and 17), and compared directly to the original baseline run with constant epsilon.

The baseline configuration (Figure 4.4) used a fixed $\epsilon = 0.7$, achieving a final best gap of 23.46%, but exhibited pronounced oscillations and large spikes throughout the 200 iterations.

The adaptive epsilon method was tested with three configurations of ϵ and α , as described previously. The corresponding results are shown in Figure 4.8–Figure 4.10, with the following best gaps recorded:

- Configuration 1: $\epsilon = 0.7, \alpha = 0.05$, best gap = 22.54%
- Configuration 2: $\epsilon = 1.0, \alpha = 0.05$, best gap = 23.18%
- Configuration 3: $\epsilon = 1.0, \alpha = 0.10$, best gap = 22.10%

A comparative analysis of the gap trajectories reveals several important trends. First, all three adaptive configurations demonstrate improved convergence stability relative to the baseline. In the constant-epsilon run (Figure 4.4), large spikes exceeding 100% are observed even in later iterations, with no clear reduction in fluctuation amplitude over time. In contrast, in the adaptive runs, while early iterations still show oscillations, there is a consistent trend towards smoother trajectories as the iterations progress.

Configuration 1 (Figure 4.8) delivers a strong overall performance. After an initially oscillatory phase, the gap progressively stabilizes, with significant reduction in spike magnitude beyond iteration 100. The final gap trajectory remains relatively stable, with minor fluctuations around the 25–30% range. The achieved best gap of 22.54% clearly outperforms the baseline value.

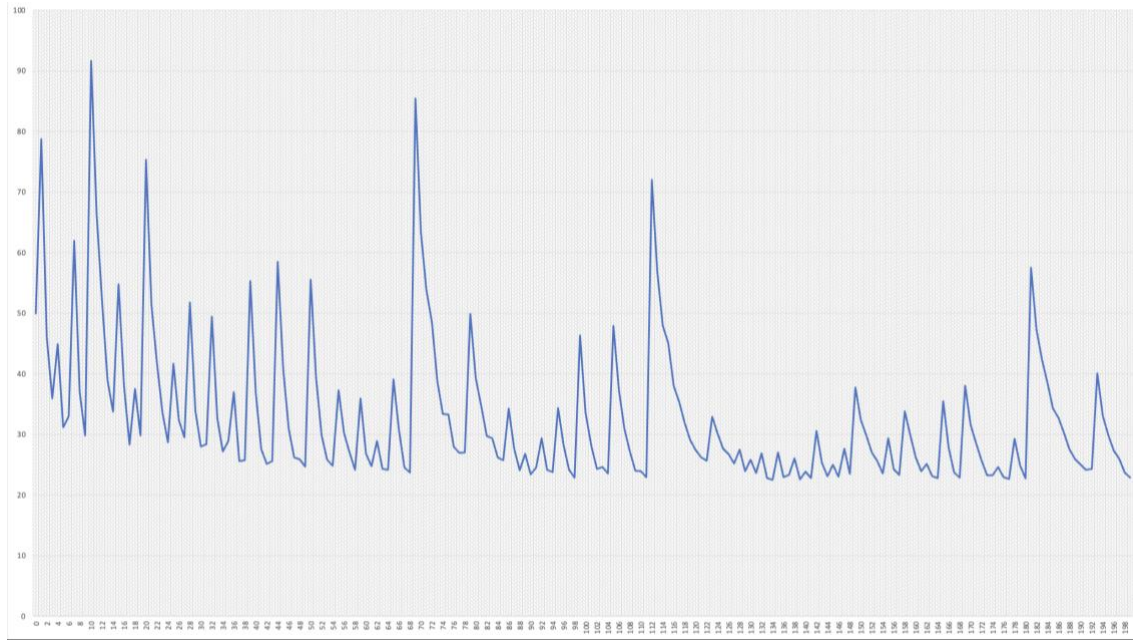


Figure 4.8 Gap Progression Plot – Scenario D, Configuration 1

Configuration 2 (Figure 4.9), using a higher starting epsilon, initially exhibits more pronounced fluctuations, including large peaks in the first 50 iterations. However, the adaptive decay mechanism effectively damps these oscillations in later iterations. While convergence stability is slightly less smooth than in Configuration 1, the final gap of 23.18% is still competitive with the baseline.

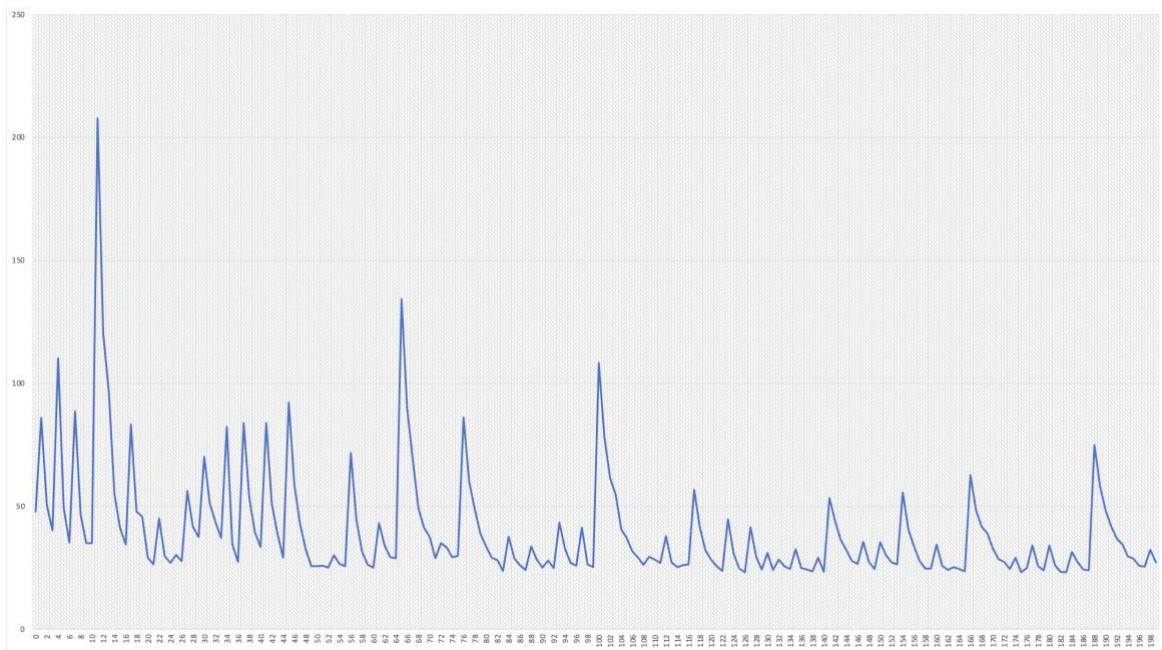


Figure 4.9 Gap Progression Plot – Scenario D, Configuration 2

Configuration 3 (Figure 4.10), with both a higher starting epsilon and a more aggressive decay rate, achieves the most stable late-stage behaviour. The oscillations are sharply reduced after iteration 100, and the final portion of the trajectory is significantly smoother than all other cases. Moreover, Configuration 3 achieves the best overall solution quality, reaching a gap of 22.10%, outperforming both the baseline and the other adaptive configurations.

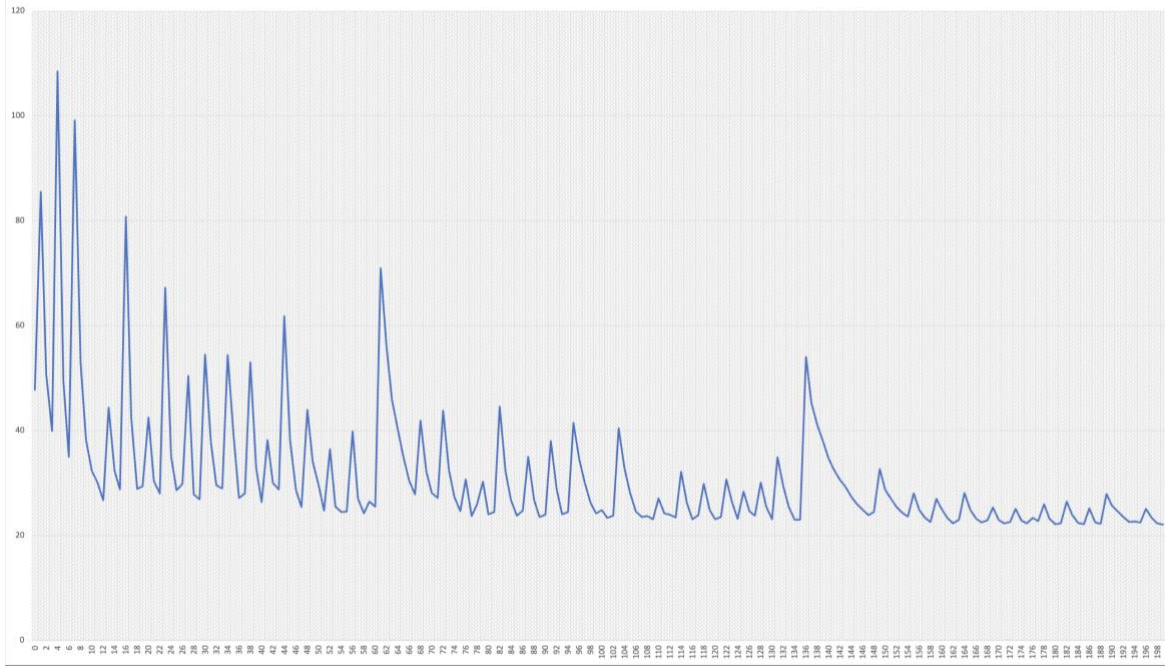


Figure 4.10 Gap Progression Plot – Scenario D, Configuration 3

In summary, the adaptive epsilon strategy leads to consistent improvements in convergence stability, with a clear reduction in oscillation amplitude over time. Furthermore, all adaptive configurations were able to surpass the best gap achieved by the constant-epsilon run. Notably, Configuration 3 demonstrates that careful tuning of the decay rate can lead to both faster stabilization and improved final solution quality.

These findings confirm the practical value of adaptive epsilon strategies in this context. By enabling larger exploratory steps in the early iterations, followed by finer adjustments as convergence is approached, the method addresses a key limitation of the constant-epsilon approach—namely, its inability to balance exploration and refinement effectively. Moreover, the adaptive method incurs no significant additional computational overhead, making it an attractive enhancement to the Lagrangian relaxation framework developed in this study.

4.6.6 Gap Evolution and Stability – Adaptive Step-Size Epsilon

In addition to the adaptive epsilon decay strategy explored in the previous section, this study further investigates an alternative approach to improving the convergence stability of the Lagrangian relaxation framework — namely, the use of adaptive step-size epsilon control based on the progress of the iterative optimization process.

Within the projected subgradient method employed for updating the Lagrange multipliers, the choice of step size parameter ϵ plays a central role in governing both convergence speed and stability. While fixed or monotonically decaying step sizes offer simplicity, they lack the flexibility to respond dynamically to the evolving optimization landscape. In particular, such approaches may either be too aggressive (causing instability and oscillations) or too conservative (leading to unnecessarily slow convergence), depending on the problem stage. To address this limitation, an adaptive step-size adjustment strategy was implemented in this work. The core idea is to enable larger steps when the optimization process is progressing effectively, and to enforce smaller, more cautious updates when nearing convergence or encountering instability — thereby providing finer control over the iterative behavior.

At each iteration k , after solving the Lagrangian dual problem, the subgradients g are computed, and the Lagrange multipliers are updated using the following standard projected subgradient update rule:

$$U^{(k+1)} = \max(0, U^k + \epsilon^k \cdot \frac{UB - z^k}{\|g\|^2} \cdot g) \quad (49)$$

where:

- UB is a known upper bound on the primal objective.
- z^k is the Lagrangian objective value at iteration k .
- g is the current subgradient vector.
- $\|g\|$ is the Euclidean norm of the subgradient.

To dynamically adapt $\epsilon^{(k)}$, the change in the Lagrangian objective is computed at each iteration as:

$$\Delta D = z^{(k)} - z^{(k-1)} \quad (50)$$

Based on this progress indicator, $\epsilon^{(k)}$ is updated according to the following rule:

$$\epsilon^{(k+1)} = \begin{cases} \min(\epsilon^{(k)} \cdot \beta_{inc}, \epsilon_{max}), & \text{if } \Delta D > \delta_{threshold} \text{ or } \|g\| < g_{threshold} \\ \max(\epsilon^{(k)} \cdot \beta_{dec}, \epsilon_{min}), & \text{Otherwise} \end{cases} \quad (51)$$

where:

- $\beta_{inc}=1.05$ is the multiplicative factor for increasing ϵ .
- $\beta_{dec}=0.80$ is the multiplicative factor for decreasing ϵ .
- $\epsilon_{min}, \epsilon_{max}$ are user-defined lower and upper bounds for ϵ .
- $\delta_{threshold}$ is the threshold for significant progress, typically set as $0.01 \times UB$.
- $g_{threshold}$ is a subgradient norm threshold, tuned empirically.

The rationale behind this mechanism is as follows: when the objective function is improving sufficiently (i.e., ΔD exceeds the threshold) or the subgradient norm becomes small (indicating proximity to optimality), ϵ is increased to encourage faster progress. Conversely, if improvement stalls or the subgradient remains large (suggesting potential instability or oscillation), ϵ is reduced to stabilize the updates and promote convergence. This adaptive step-size strategy is inspired by classical techniques for tuning subgradient methods in nonsmooth optimization — particularly the family of Polyak-inspired adaptive step size approaches — but has been tailored here to reflect the dynamics of the large-scale, highly constrained maintenance scheduling problem under study. By allowing the algorithm to automatically adjust its aggressiveness based on observed progress, this approach aims to provide a more robust and effective means of controlling convergence behavior, especially in scenarios where fixed or decaying step sizes struggle to balance exploration and refinement.

In the following section, the results of applying this adaptive step-size strategy to Scenario D will be presented and analyzed, with a direct comparison to both the baseline constant-epsilon run and the previously tested adaptive decay strategy.

Adaptive Step-Size Epsilon Results

Figure 4.11 presents the gap evolution when applying the Adaptive Step-Size Epsilon strategy, as introduced in the previous section, to Scenario D. For reference, the best gap achieved in this run was 21.41%, an improvement over both the baseline constant-epsilon run (*best gap* 23.46%) and the decaying-epsilon scenarios tested earlier (*best gaps* 22.1% to 23.18%).

In terms of convergence behavior, the adaptive step-size approach exhibits substantially improved stability compared to both the constant-epsilon and simple decay-based strategies. While the initial iterations (up to approximately iteration 50) still display oscillations — expected given the large step sizes favored early in the adaptive scheme — the amplitude of these oscillations diminishes progressively as the algorithm adjusts to smaller step sizes. Beyond iteration 100, the gap trajectory becomes markedly smoother, with no major spikes observed after this point, and demonstrates a consistent trend toward convergence. Notably, in contrast to the constant-epsilon baseline (Figure 4.4), which exhibited recurring large spikes even in late iterations, the adaptive approach effectively suppresses such instability. Similarly, compared to the simple decaying-epsilon scenarios, the adaptive step-size leads to faster stabilization and a lower final gap.

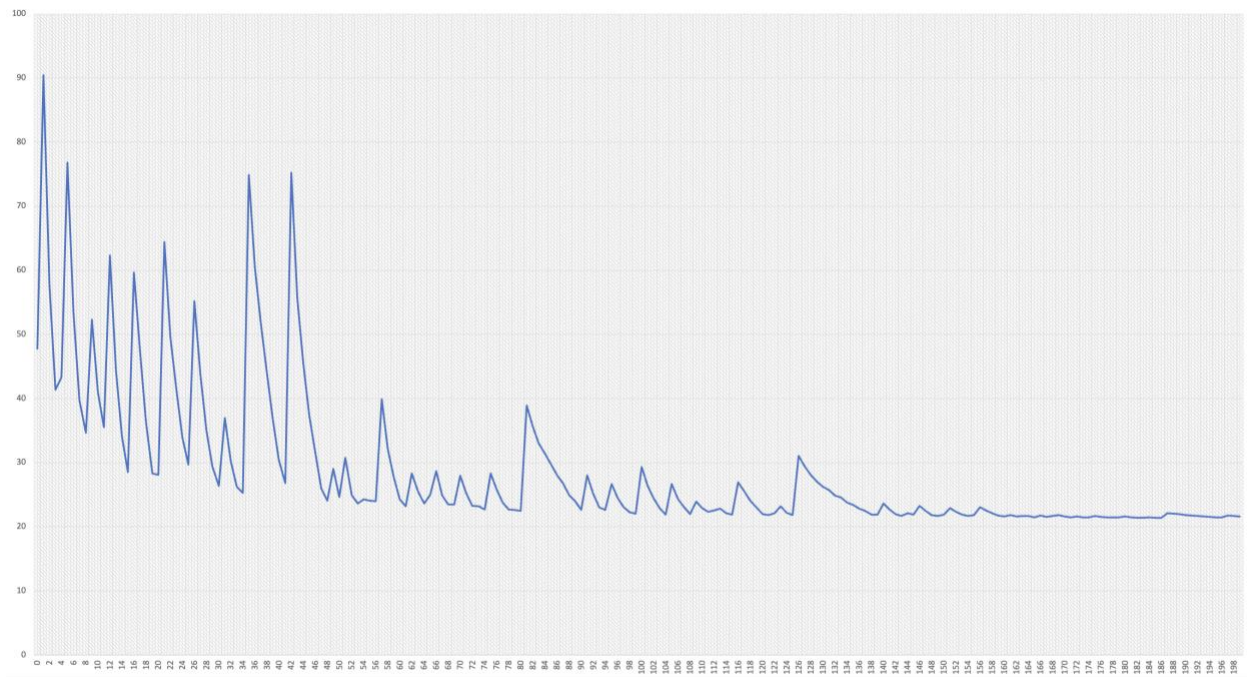


Figure 4.11 Gap Progression Plot – Scenario D, Adaptive Step-Size Epsilon

These results suggest that the Adaptive Step-Size Epsilon mechanism not only enhances the convergence stability of the Lagrangian relaxation but also contributes to achieving better solution quality. The empirical evidence here aligns well with prior findings in the literature, such as those reported by [\(Bento et al., 2020\)](#), where adaptive control of Lagrange multiplier updates was shown to significantly improve performance in large-scale, nonsmooth optimization problems.

Importantly, this improvement in convergence behavior and gap quality comes with no significant increase in computational cost. The runtime associated with the adaptive strategy was comparable to that of the constant and decaying epsilon methods, indicating that the added robustness and precision do not require additional computational effort. Based on these promising results, the Adaptive Step-Size Epsilon approach represents a valuable enhancement to the Lagrangian framework employed in this study, and it will be carried forward into subsequent computational experiments on larger problem instances.

4.6.7 Large-Scale Instance Testing: Scenario Selection and Rationale

In this section, the Lagrangian relaxation framework is applied to large-scale problem instances to assess its scalability and solution behavior. The selection of the appropriate relaxation scenario for large-scale testing is a critical step, as the computational cost increases substantially with instance size. Several factors informed the selection.

First, scenarios relaxing a single constraint (Scenarios A, B, and C) were observed to be computationally expensive even on small instances, without yielding sufficient improvement in gap performance (except Scenario 3 which still requires a long runtime) to justify their use in large-scale tests.

Second, Scenario G, involving full relaxation of all three constraints, produced poor solution quality on small instances (best gap of 31.11%), making it unsuitable for large-scale testing.

Third, Scenario F (relaxing Constraints 16 and 17) achieved the best solution quality among the combined relaxation scenarios on small instances (best gap of 14.47%). However, this was obtained at a significantly higher computational cost (4162.4 seconds), raising concerns about feasibility on large-scale instances. By contrast, Scenario D (relaxing Constraints 15 and 17) offers

a compelling compromise. It produced the second-best solution quality (best gap of 23.46% with constant epsilon, improved to 21.41% with adaptive epsilon) while achieving the shortest computational time (817.2 seconds) among the combined-relaxation scenarios. Furthermore, previous analysis showed that its gap evolution stability improved markedly with the adaptive step-size epsilon strategy, providing further confidence in its robustness for larger instances.

Based on these considerations, Scenario D — with adaptive step-size epsilon — is selected as the most suitable configuration for evaluating the Lagrangian relaxation framework on large-scale problem instances. The following section presents and analyzes the results of this experiment.

4.6.8 Results on Large-Scale Instance

To assess the practical scalability of the proposed Lagrangian relaxation approach, the selected configuration—Scenario D—was applied to the large-scale instance introduced in Chapter 3. This test aims to evaluate the performance of the algorithm under significantly increased problem size and complexity, with particular attention to the gap evolution behavior and final solution quality. Two versions of the algorithm were tested: one with a fixed epsilon value ($\varepsilon = 0.7$) and another using the adaptive step-size epsilon adjustment strategy previously described. Both were executed under equivalent computational settings, and the performance was evaluated across 95 iterations to allow for direct comparison.

The fixed epsilon run resulted in a best gap of 52.13%, with notable volatility throughout the process (see Figure 4.12). Although the method steadily reduced the gap over time, frequent spikes and regressions in gap values were observed. These fluctuations indicate instability in the convergence process, a known limitation of constant step-size subgradient methods, particularly in high-dimensional spaces with intricate constraint interactions. Beyond the 90th iteration, the improvements diminished considerably, and further iterations did not meaningfully reduce the gap.

In contrast, the adaptive epsilon version achieved a best gap of 33.01% within the same 95-iteration horizon—a substantial improvement of nearly 20 percentage points. Furthermore, the trajectory of gap values exhibited a markedly smoother descent, with fewer sharp fluctuations and more consistent progress toward convergence (Figure 4.13). When the run was extended to 200 iterations, the gap further improved to 31.61%, though the marginal gains diminished significantly

after iteration 100. This plateau suggests that most of the improvement in solution quality occurs within the early to mid stages of the iteration process, validating the effectiveness of the adaptive epsilon in accelerating convergence.

A summary comparison of the results is provided in Table 4.3. Notably, the adaptive strategy not only outperforms the constant epsilon approach in final solution quality but also exhibits more stable and efficient convergence behavior. This confirms the earlier findings on small instances and demonstrates that the benefits of adaptive control mechanisms extend to large-scale implementations as well.

Table 4.3 Performance Comparison on Large Instance (Scenario D)

Strategy	Best Gap (%) until 95th iteration	Iteration Best Gap	Final Iteration Gap (%)	Fluctuation Level	Remarks
Constant Epsilon ($\varepsilon=0.7$)	52.13	93	52.13	High	Unstable; limited gain after 90 iterations
Adaptive Epsilon	33.01	92	31.61	Low	Stable; improved early convergence

Figure 4.12 and Figure 4.13 illustrate the gap trajectories for the fixed and adaptive epsilon implementations, respectively. The full iteration-level results for the adaptive epsilon is included in Appendix B for transparency and reproducibility.

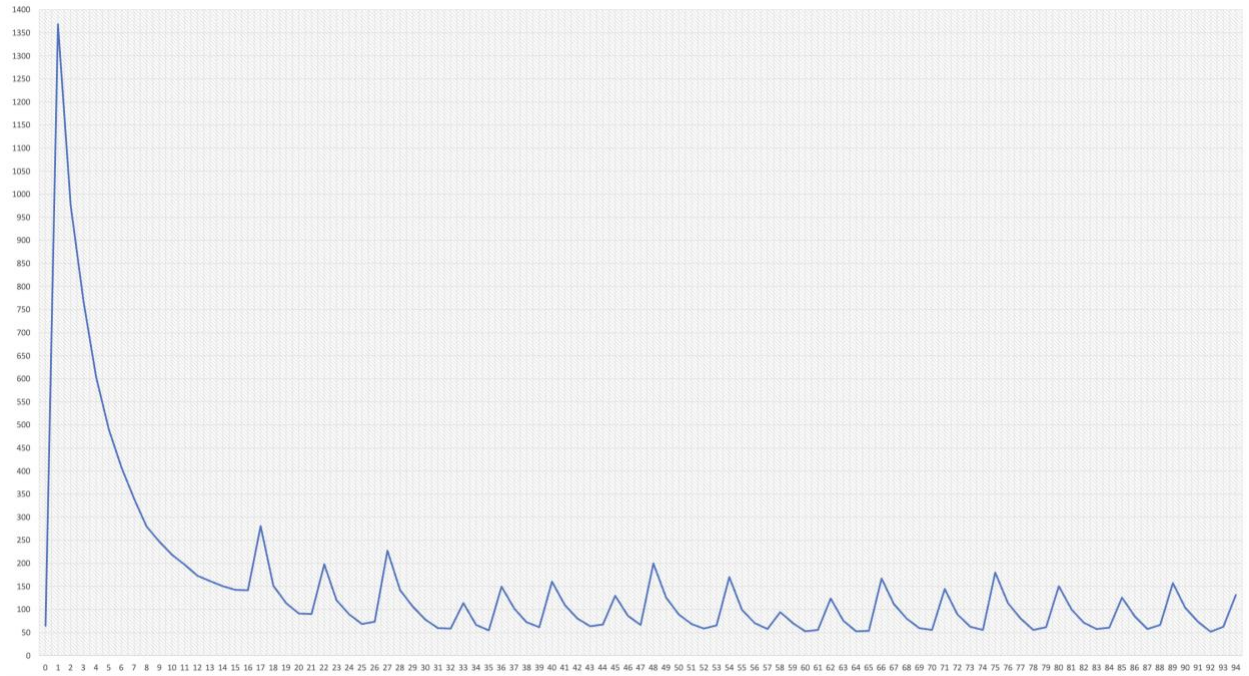


Figure 4.12 Gap Progression Plot – Scenario D, Large Instance – Constant Epsilon

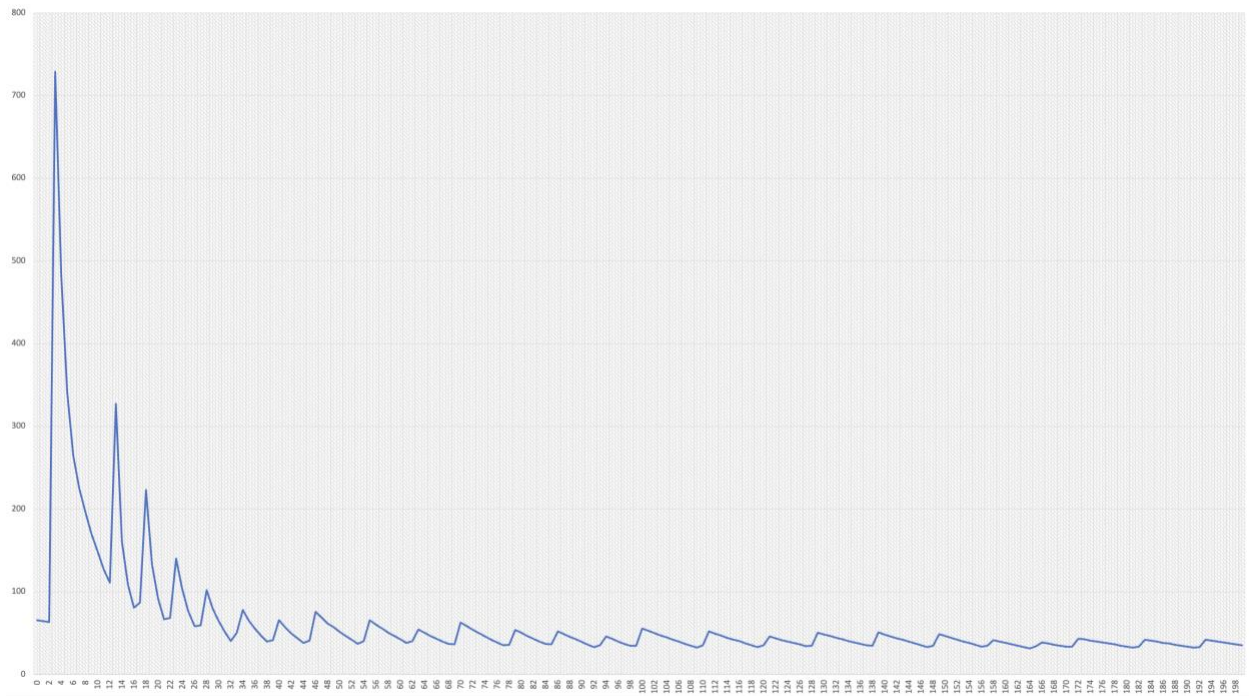


Figure 4.13 Gap Progression Plot – Scenario D, Large Instance – Adaptive Epsilon

Taken together, these results reinforce the effectiveness of the adaptive step-size approach in stabilizing and accelerating convergence, even in the context of complex, large-scale MRO scheduling problems. The improved gap values also suggest that the method can provide viable

lower bounds for subsequent heuristic or metaheuristic solution strategies in integrated solution frameworks.

4.7 Conclusion

This chapter explored the development and evaluation of a Lagrangian relaxation framework tailored for operational-level MRO scheduling with remanufacturing. The primary objective was to manage the complexity introduced by interdependent constraints and remanufacturing decisions, while generating meaningful lower bounds for subsequent heuristic-based improvements. Through a structured relaxation of selected constraints, paired with subgradient optimization, the chapter presented both a methodological foundation and a detailed computational study on the effectiveness of this approach.

The initial analysis focused on identifying the impact of relaxing different combinations of constraints. Among the seven Scenarios tested, Scenario D—relaxing resource constraints and Component dependencies—emerged as the most promising candidate due to its balance between solution quality and computational efficiency. While Scenario F achieved a marginally better gap, its significantly higher runtime rendered it less practical for integration into real-time or near-real-time decision environments.

A detailed investigation into gap evolution trajectories across all scenarios revealed critical insights into convergence behavior. Scenarios that involved relaxation of only one constraint (e.g., Scenario 1) exhibited smoother convergence profiles, but their solution quality was often suboptimal. In contrast, multi-constraint relaxations provided better solution quality but at the expense of increased oscillation and instability. Scenario D, while affected by these issues, still maintained relatively lower fluctuations than other multi-constraint settings, validating its selection for further enhancement.

To improve the convergence behavior, the chapter introduced and evaluated two adaptive epsilon adjustment mechanisms in the subgradient optimization process. The first employed a deterministic decay rule, which led to significant improvements in gap stability and marginal improvements in solution quality over the fixed-epsilon baseline. The second, inspired by (Bento et al., 2020), implemented a more responsive adaptive step-size strategy that adjusted epsilon

based on changes in objective value and gradient norm. This method achieved the most stable convergence and yielded the lowest gap among all tested configurations, demonstrating its superiority in balancing step-size aggressiveness and stability across iterations.

The scalability of the framework was further validated by applying Scenario D, enhanced with adaptive step-size control, to the large-scale problem instance introduced earlier in the thesis. The method maintained its computational tractability and continued to exhibit stable convergence characteristics. The best gap achieved in this context was *31.61%*, a notable improvement over the fixed-epsilon result of *52.13%*. Additionally, diminishing returns beyond the 100th iteration emphasized the efficiency of the adaptive approach in achieving rapid early convergence.

Overall, this chapter not only established the feasibility of Lagrangian relaxation in the context of complex MRO scheduling with remanufacturing but also demonstrated how convergence behavior and solution quality can be substantially enhanced through carefully designed adaptive step-size strategies. The insights gained here lay a solid foundation for integrating Lagrangian bounds into hybrid optimization frameworks and pave the way for further exploration into decomposition-based methods and coordination with heuristic improvement phases.

Chapter 5

Conclusion and Future Work

This thesis investigated a complex real-world problem in the domain of aviation maintenance: the operational-level scheduling of aircraft maintenance checks. We proposed a novel modeling and solution framework that balances optimization accuracy with computational tractability. Through a sequence of progressively refined models and algorithms, we explored both heuristic and exact approaches, culminating in a hybrid framework that integrates iterative local search and Lagrangian relaxation. This final chapter provides an overview of each chapter’s contributions and outlines potential directions for future research.

5.1 Summary of Contributions

Chapter 1 introduced the research problem, its significance in the aviation industry, and the key challenges posed by the highly constrained and large-scale nature of maintenance scheduling. A detailed literature review positioned our work within the context of existing MRO optimization research, highlighting the need for more scalable, operationally grounded models and methods.

Chapter 2 presented the baseline mathematical formulation of the problem as a mixed-integer linear program (MILP), capturing task durations, resource constraints, sequencing dependencies, and component relationships. This chapter laid the foundation for formal modeling and emphasized the problem’s computational difficulty, especially for real-world-sized instances. The formulation also introduced the fundamental sets, parameters, and constraints that would be adapted in later chapters.

Chapter 3 proposed an Iterative Local Search (ILS) metaheuristic designed to efficiently generate high-quality feasible schedules. The ILS method incorporated tailored neighborhood operators, local improvement strategies, and restart mechanisms. Experimental results demonstrated that the ILS could reliably generate near-optimal solutions for both small and large problem instances, serving not only as a practical heuristic but also as a source of upper bounds for the Lagrangian framework introduced in the next chapter.

Chapter 4 explored the use of Lagrangian Relaxation (LR) to compute lower bounds for the scheduling problem. By relaxing various combinations of resource, dependency, and sequencing constraints, we decomposed the original MILP into more tractable subproblems. We implemented a projected subgradient optimization algorithm in Python to solve the Lagrangian duals, analyzing convergence stability and bound quality across different relaxation schemes. This chapter also evaluated adaptive step-size strategies for updating multipliers, including both decay-based and progress-based adaptive methods. The results demonstrated that adaptive strategies could significantly improve stability without increasing computational effort, especially in large-scale settings.

Together, these chapters presented a comprehensive hybrid framework for MRO scheduling: a high-quality heuristic for producing feasible solutions (upper bounds), and a rigorous relaxation method for computing lower bounds and assessing optimality gaps.

5.2 Recommendations for Future Work

Although this thesis provides a strong foundation for operational-level maintenance scheduling, several research opportunities remain:

1. **Advanced Primal Heuristics from Dual Solutions:** In the current framework, feasible solutions are generated solely by the ILS heuristic. Future work could investigate dual-to-primal heuristics that leverage the structure of relaxed solutions (e.g., repair-based strategies or randomized rounding) to construct feasible schedules directly from the Lagrangian dual.
2. **Improved Multiplier Update Methods:** While the subgradient method was effective, convergence can be slow and sensitive to step-size parameters. Future research could explore more advanced methods such as bundle methods, quasi-Newton updates, or cutting-plane techniques to solve the dual more efficiently.
3. **Stochastic and Robust Optimization:** In practice, many parameters (e.g., task durations, resource availability) are uncertain. Extending the model to consider stochastic or robust formulations could improve schedule resilience in the face of disruption.
4. **Parallel and Distributed Solvers:** For extremely large instances, solving Lagrangian subproblems in parallel could dramatically reduce computation time. Investigating

distributed or GPU-accelerated solvers for MILP subproblems is another avenue worth pursuing.

5. **Interactive Decision Support Tools:** Finally, embedding the proposed model and algorithms into a real-time decision support interface for maintenance planners could facilitate adoption in industry. Visualization, human-in-the-loop overrides, and scenario analysis tools would enhance usability and trust.

5.3 Closing Remarks

This thesis contributed to the growing field of optimization for complex maintenance operations by providing both theoretical insights and practical algorithms for scheduling aircraft checks. The combination of mathematical rigor, heuristic flexibility, and computational experimentation offers a robust foundation for future academic exploration and industrial application. By bridging the gap between exact methods and heuristic efficiency, this research advances the state of the art in MRO optimization and paves the way for scalable, intelligent maintenance planning systems.

Appendix A

Parameters and notations used in the cost calculation section:

CPN : Set of components

TSK : Set of tasks

C_{jr} : Cost of performing task r on component j in the current maintenance window

LC_{jr} : The total labor cost for task r in component j

MC_{jr} : The total machine cost for task r in component j

PC_{jr} : The total material cost for task r in component j

LU_{jrk} : The number of labors needed for activity k in task r in component j

MU_{jrk} : The number of machines needed for activity k in task r in component j

D_{jrk} : The duration for activity k of task r in component j

HLC : Hourly labor cost

HMC : Hourly machine cost

ECN_{jr} : Estimated cost of the next maintenance for component j if maintenance task r is performed

CN_{jr} : Cost for component j in the future maintenance window if we perform task r on it no

PN_{jr} : Probability of not failing before the next maintenance for component j if task r done

ECU_j : Estimated cost of unplanned maintenance of the component j

CU_j : Cost of unplanned maintenance for component j

PU_{jr} : Probability of failing before the next maintenance for component j if task r performed

FD_j : The total flight delay cost caused by an unplanned maintenance of component j

DOS_j : The time that airplane is out of service due to unplanned maintenance of component j

HFD : Flight delay cost for each hour caused by unplanned maintenance

TT : Transport time for moving airplane to the maintenance facility

ST : Slack time

NS : Number of Seats

SP : Price of a seat for a day

FP : The percentage of the seats that is expected to be full for the flight

Appendix B

Table B.1 – Full Iteration Results on the large-scale problem with adaptive step-size

Iteration	Epsilon	Objective	Gap	Time
0	1	270053.77	65.7	390.1
1	0.8	281234.28	64.28	5548.2
2	0.84	287942.93	63.42	7465.5
3	0.882	-4952363.2	729.09	7794.1
4	0.7056	-3017267.6	483.28	8105.7
5	0.74088	-1896637.2	340.92	8613.3
6	0.777924	-1296037.6	264.63	9293.7
7	0.8168202	-980519.97	224.55	9978
8	0.85766121	-761483.78	196.73	11461.8
9	0.90054427	-554393.09	170.42	15906.1
10	0.94557148	-390372.9	149.59	17822.9
11	0.99285006	-208733.25	126.51	19099.1
12	1.04249256	-86509.3	110.99	20330.7
13	1.09461719	-1792301.2	327.67	20885.3
14	0.87569375	-477556.66	160.66	21459.2
15	0.91947844	-67623.11	108.59	22639.8
16	0.96545236	153203.46	80.54	23947.6
17	1.01372498	101427.27	87.12	25196.2
18	0.81097998	-967372.07	222.88	25798.6
19	0.64878399	-256835.51	132.63	26416
20	0.68122319	65409.67	91.69	27039.5
21	0.71528435	261659.4	66.76	29258.5
22	0.75104856	249090.02	68.36	55590
23	0.60083885	-316068.04	140.15	56294.2
24	0.48067108	-28010.43	103.56	57014.9
25	0.50470463	182195.67	76.86	58359.1
26	0.52993987	326873.11	58.48	60077.6
27	0.55643686	318369.23	59.56	61284.6
28	0.44514949	-17733.46	102.25	62055.7
29	0.35611959	160480.02	79.61	62836.3
30	0.37392557	277563.17	64.74	63980.2
31	0.39262185	379993.81	51.73	65605
32	0.41225294	468607.15	40.47	66925.7

Table B.2 – Full Iteration Results on the large-scale problem with adaptive step-size (Continued)

33	0.43286559	388825.8	50.61	68398.2
34	0.34629247	172225.07	78.12	69234.6
35	0.27703398	276244.58	64.91	70303.6
36	0.29088567	351029.04	55.41	71928.4
37	0.30542996	415613.03	47.21	74593.8
38	0.32070146	474740.32	39.69	75968.6
39	0.33673653	462012.97	41.31	77358.8
40	0.26938922	270166.92	65.68	78111.4
41	0.21551138	342223.28	56.53	79255.9
42	0.22628695	396737.16	49.6	80746.3
43	0.23760129	442700.73	43.76	82395.9
44	0.24948136	486033.21	38.26	83984.3
45	0.26195543	463451.43	41.13	85569
46	0.20956434	189865.55	75.88	86367.6
47	0.16765147	243496.71	69.07	87256.6
48	0.17603405	301720.07	61.67	88147
49	0.18483575	335843.8	57.34	89251.7
50	0.19407754	383277.73	51.31	90569.5
51	0.20378141	421232.23	46.49	92597.6
52	0.21397048	456959.97	41.95	94216.1
53	0.22466901	494223.79	37.22	100640.4
54	0.23590246	471407.08	40.12	105612.3
55	0.18872197	272620.9	65.37	106615.8
56	0.15097757	315419.61	59.93	107483.2
57	0.15852645	351428.1	55.36	108747.9
58	0.16645278	388880.38	50.6	110298.1
59	0.17477541	421497.07	46.46	111964.3
60	0.18351418	451947.83	42.59	117676.7
61	0.19268989	486481.51	38.2	119717.7
62	0.20232439	468031.94	40.55	139379.5
63	0.16185951	359143.06	54.38	140784.2
64	0.12948761	389533.29	50.52	142195.5
65	0.13596199	419357.67	46.73	143909.5
66	0.14276009	446660.05	43.26	145528.3
67	0.14989809	473366.58	39.87	147852.4
68	0.157393	495900.35	37.01	149222.9
69	0.16526265	498359.94	36.69	150835.4

Table B.3 – Full Iteration Results on the large-scale problem with adaptive step-size (Continued)

70	0.17352578	292734.88	62.81	151875.4
71	0.13882062	329596.27	58.13	152915.1
72	0.14576166	363288.82	53.85	154297
73	0.15304974	394705.58	49.86	155819
74	0.16070222	425749.43	45.92	157496
75	0.16873734	455016.62	42.2	159879.2
76	0.1771742	482641.16	38.69	162108.5
77	0.18603291	509343.04	35.3	163750.1
78	0.19533456	505470.11	35.79	165647.1
79	0.15626765	362312.38	53.98	167035.3
80	0.12501412	391187.87	50.31	168592.4
81	0.13126482	420513.27	46.58	170392.5
82	0.13782806	445841.38	43.37	172372.9
83	0.14471947	474850.62	39.68	174240.9
84	0.15195544	495203.77	37.1	175947.2
85	0.15955321	501755.12	36.26	178957.6
86	0.16753087	374943.49	52.37	180719.5
87	0.1340247	402146.35	48.92	182510.5
88	0.14072593	428432.25	45.58	184525.8
89	0.14776223	452474.82	42.52	190035.3
90	0.15515034	477386.66	39.36	191706.5
91	0.16290786	504069.11	35.97	194817.1
92	0.17105325	527371.8	33.01	197289.2
93	0.17960592	507194.66	35.57	199322.2
94	0.14368473	426785.75	45.79	201221.7
95	0.11494779	448390.06	43.04	203303.8
96	0.12069517	472172.27	40.02	211229
97	0.12672993	493943.34	37.26	224867.5
98	0.13306643	514619.23	34.63	227341.2
99	0.13971975	513590.5	34.76	229100.5
100	0.1117758	351554.56	55.34	230636.5
101	0.08942064	373054.15	52.61	232227.8
102	0.09389167	394623.19	49.87	234146.6
103	0.09858626	414607.03	47.33	236201.5
104	0.10351557	435983.68	44.62	238364.9
105	0.10869135	456047.24	42.07	240463.2
106	0.11412592	475302.03	39.62	242293.2

Table B.4 – Full Iteration Results on the large-scale problem with adaptive step-size (Continued)

107	0.11983221	495716.72	37.03	245145.6
108	0.12582382	514905.65	34.59	246938.7
109	0.13211501	532393.4	32.37	248873.4
110	0.13872076	508529.48	35.4	251515.1
111	0.11097661	377616.42	52.03	253089.9
112	0.08878129	398378.65	49.4	255160.3
113	0.09322035	418687.66	46.82	257170.6
114	0.09788137	436971.53	44.49	259183.9
115	0.10277544	455087.5	42.19	261036.5
116	0.10791421	470775.58	40.2	274173.8
117	0.11330992	489803.45	37.78	276645.7
118	0.11897542	507489.45	35.53	280737
119	0.12492419	526587.99	33.11	285613
120	0.1311704	510889.7	35.1	297661.6
121	0.10493632	425240.64	45.98	299916.7
122	0.08394906	441971.96	43.86	302575
123	0.08814651	459359.22	41.65	305317.1
124	0.09255383	473345.67	39.87	315801.7
125	0.09718152	489183.03	37.86	317807.5
126	0.1020406	502172.61	36.21	319703
127	0.10714263	517463.4	34.27	321578.6
128	0.11249976	514572.9	34.64	324040.3
129	0.08999981	389000.7	50.59	326071
130	0.07199985	405833.79	48.45	327997.8
131	0.07559984	420691.57	46.56	330218.9
132	0.07937983	436800.32	44.51	332480.1
133	0.08334882	451866.97	42.6	334717.3
134	0.08751627	467578.63	40.6	337118.4
135	0.09189208	482068.14	38.76	340110.1
136	0.09648668	495111.45	37.11	342527.9
137	0.10131102	509155.64	35.32	344633.1
138	0.10637657	513486.43	34.77	350269.9
139	0.1116954	386361.14	50.92	352437.1
140	0.08935632	406619.41	48.35	354617.7
141	0.09382413	425329.48	45.97	356869
142	0.09851534	443174.11	43.7	359448.5
143	0.10344111	457052.95	41.94	365121.4

Table B.5 – Full Iteration Results on the large-scale problem with adaptive step-size (Continued)

144	0.10861316	475405.59	39.61	369236.4
145	0.11404382	492527.59	37.44	377522
146	0.11974601	510864.25	35.11	401328.1
147	0.12573331	526494.85	33.12	404700.4
148	0.13201998	514298.05	34.67	407842
149	0.10561598	404668.25	48.6	410365.6
150	0.08449279	421979.42	46.4	412777.7
151	0.08871742	440257.66	44.08	415291.1
152	0.0931533	456127.43	42.06	418144.1
153	0.09781096	472006.97	40.04	419891
154	0.10270151	488285.65	37.97	421657.9
155	0.10783658	503607.15	36.03	423222.7
156	0.11322841	521810.95	33.72	429501.2
157	0.11888983	514209.02	34.68	431491.7
158	0.09511187	462787.82	41.21	434528.8
159	0.07608949	475653.16	39.58	437879.6
160	0.07989397	488512.86	37.95	443432
161	0.08388867	502282.47	36.2	446027
162	0.0880831	512772.18	34.86	448811.5
163	0.09248725	526546.68	33.11	452194.4
164	0.09711162	538418.83	31.61	455472.5
165	0.1019672	517582.98	34.25	457574.6
166	0.08157376	480913.42	38.91	461402.2
167	0.06525901	493115.3	37.36	465844.7
168	0.06852196	502784.6	36.13	488767.4
169	0.07194806	513638.58	34.75	491982.1
170	0.07554546	524049.5	33.43	493805.1
171	0.07932273	523335.81	33.52	497236.8
172	0.06345818	445727.3	43.38	498330.2
173	0.05076655	453148.06	42.44	501338.6
174	0.05330488	462917.64	41.2	504217.2
175	0.05597012	473710.74	39.83	507020.5
176	0.05876862	482694.49	38.68	511605
177	0.06170706	493128.35	37.36	514742.3
178	0.06479241	502188.67	36.21	523356.5
179	0.06803203	511377.3	35.04	525413.9
180	0.07143363	521543.27	33.75	527006.5

Table B.6 – Full Iteration Results on the large-scale problem with adaptive step-size (Continued)

181	0.07500531	532295.51	32.38	530783.5
182	0.07875558	520623.02	33.87	538398.5
183	0.06300446	455540.31	42.13	541462.8
184	0.05040357	465354.65	40.89	542952.8
185	0.05292375	475039.65	39.66	545937.5
186	0.05556994	486301.61	38.23	546248.3
187	0.05834843	493000.01	37.38	548439.6
188	0.06126585	502630.79	36.15	550389.2
189	0.06432915	511535.33	35.02	576797.9
190	0.0675456	520799.73	33.84	580154.5
191	0.07092288	529578.27	32.73	584598.8
192	0.07446903	527865.26	32.95	588352.6
193	0.05957522	455463.3	42.14	590137.5
194	0.04766018	465019.26	40.93	591521.1
195	0.05004319	473703.28	39.83	593567.6
196	0.05254535	482954.45	38.65	595798.5
197	0.05517261	490530.9	37.69	597484
198	0.05793124	499600.63	36.54	601809.3

Biography

Ahmed, A. A. A., Dwijendra, N. K. A., Bynagari, N., Modenov, A. K., Kavitha, M., &

Dudukalov, E. (2021). Multi Project Scheduling and Material Planning Using Lagrangian Relaxation Algorithm. *Industrial Engineering & Management Systems*, 20(4), 580–587.
<https://doi.org/10.7232/iems.2021.20.4.580>

Antonakis, A. S., & Giannakoglou, K. C. (2018). Optimisation of military aircraft engine maintenance subject to engine part shortages using asynchronous metamodel-assisted particle swarm optimisation and Monte-Carlo simulations. *International Journal of Systems Science: Operations & Logistics*, 5(3), 239–252.
<https://doi.org/10.1080/23302674.2017.1289421>

Archetti, C., Peirano, L., & Speranza, M. G. (2022). Optimization in multimodal freight transportation problems: A Survey. *European Journal of Operational Research*, 299(1), 1–20.

Asmatulu, E., Twomey, J., & Overcash, M. (2013). Evaluation of recycling efforts of aircraft companies in Wichita. *Resources, Conservation and Recycling*, 80, 36–45.

Bento, P., Pina, F., Mariano, S., & Do Rosario Calado, M. (2020). Short-term Hydro-thermal Coordination By Lagrangian Relaxation: A New Algorithm for the Solution of the Dual Problem. *KnE Engineering*. <https://doi.org/10.18502/keg.v5i6.7093>

Boeing. (2023). *Commercial Market Outlook 2023–2042*.
<https://www.boeing.com/content/theboeingcompany/us/en/commercial/market/commercial-market-outlook>

- Boere, N. J. (1977). Air Canada Saves with Aircraft Maintenance Scheduling. *Interfaces*, 7(3), 1–13. <https://doi.org/10.1287/inte.7.3.1>
- Bragin, M. A. (2023). *Survey on Lagrangian Relaxation for MILP: Importance, Challenges, Historical Review, Recent Advancements, and Opportunities* (No. arXiv:2301.00573). arXiv. <https://doi.org/10.48550/arXiv.2301.00573>
- Bragin, M. A., Luh, P. B., Yan, B., & Sun, X. (2018). A scalable solution methodology for mixed-integer linear programming problems arising in automation. *IEEE Transactions on Automation Science and Engineering*, 16(2), 531–541.
- Bragin, M. A., Luh, P. B., Yan, J. H., Yu, N., & Stern, G. A. (2015). Convergence of the Surrogate Lagrangian Relaxation Method. *Journal of Optimization Theory and Applications*, 164(1), 173–201. <https://doi.org/10.1007/s10957-014-0561-3>
- Bragin, M. A., & Tucker, E. L. (2022). Surrogate “Level-Based” Lagrangian Relaxation for mixed-integer linear programming. *Scientific Reports*, 12(1), 22417. <https://doi.org/10.1038/s41598-022-26264-1>
- Bragin, M. A., Yan, B., & Luh, P. B. (2020). Distributed and asynchronous coordination of a mixed-integer linear system via surrogate Lagrangian relaxation. *IEEE Transactions on Automation Science and Engineering*, 18(3), 1191–1205.
- Bulbul, K. G., & Kasimbeyli, R. (2021). Augmented Lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem. *Computers & Operations Research*, 132, 105294. <https://doi.org/10.1016/j.cor.2021.105294>
- Bureau of Transportation Statistics. (2017). *Airline On-Time Performance and Causes of Flight Delays* | Bureau of Transportation Statistics. U.S. Department of Transportation.

<https://www.bts.gov/explore-topics-and-geography/topics/airline-time-performance-and-causes-flight-delays>

Cook, A. J., Tanner, G., & Anderson, S. (2004). *Evaluating the true cost to airlines of one minute of airborne or ground delay*.

<https://westminsterresearch.westminster.ac.uk/item/931w2/evaluating-the-true-cost-to-airlines-of-one-minute-of-airborne-or-ground-delay-final-report>

Cook, A. J., Tanner, G., & Lawes, A. (2012). The hidden cost of airline unpunctuality. *Journal of Transport Economics and Policy*, 46(2), Article 2.

Cooper, T., Smiley, J., Porter, C., & Precourt, C. (2018). Global fleet & MRO market forecast commentary. *Olyver Wyman*. http://opweb-bucket-prod.s3.amazonaws.com/pdf/16479211930482018-2028_Global_Fleet_MRO_Market_Forecast_Commentary_Public_Final_web.pdf

Dastgoshade, S., Abraham, A., & Fozooni, N. (2021). The Lagrangian Relaxation Approach for Home Health Care Problems. In A. Abraham, Y. Ohsawa, N. Gandhi, M. A. Jabbar, A. Haqiq, S. McLoone, & B. Issac (Eds.), *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)* (pp. 333–344). Springer International Publishing. https://doi.org/10.1007/978-3-030-73689-7_32

Demirel, N. Ö., & Gökçen, H. (2008). A mixed integer programming model for remanufacturing in reverse logistics environment. *The International Journal of Advanced Manufacturing Technology*, 39(11–12), 1197–1206. <https://doi.org/10.1007/s00170-007-1290-7>

Deng, Q., Santos, B. F., & Curran, R. (2020). A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research*, 281(2), 256–273. <https://doi.org/10.1016/j.ejor.2019.08.025>

- Deng, Q., Santos, B. F., & Verhagen, W. J. (2021). A novel decision support system for optimizing aircraft maintenance check schedule and task allocation. *Decision Support Systems*, 146, 113545.
- Erlenkotter, D. (1978). A Dual-Based Procedure for Uncapacitated Facility Location. *Operations Research*, 26(6), 992–1009. <https://doi.org/10.1287/opre.26.6.992>
- Ertogral, K., & Öztürk, F. S. (2019). An integrated production scheduling and workforce capacity planning model for the maintenance and repair operations in airline industry. *Computers & Industrial Engineering*, 127, 832–840. <https://doi.org/10.1016/j.cie.2018.11.022>
- Ferrer, G., Heath, S. K., & Dew, N. (2011). An RFID application in large job shop remanufacturing operations. *International Journal of Production Economics*, 133(2), 612–621.
- Fisher, M. L., & Shapiro, J. F. (1974). Constructive Duality in Integer Programming. *SIAM Journal on Applied Mathematics*, 27(1), 31–52. <https://doi.org/10.1137/0127003>
- Ghamary, Y., & Chen, M. (2024). A mathematical model for aerospace product MRO scheduling with remanufacturing. *Journal of Remanufacturing*, 14(1), 93–123. <https://doi.org/10.1007/s13243-024-00135-6>
- Goffin, J. L. (1977). On convergence rates of subgradient optimization methods. *Mathematical Programming*, 13(1), 329–347. <https://doi.org/10.1007/BF01584346>
- Goffin, J.-L., & Kiwiel, K. C. (1998). *Convergence of a simple subgradient level method*. Groupe d'études et de recherche en analyse des décisions. <https://www.academia.edu/download/52172264/s10107005005320170315-1954-10irpwc.pdf>

- Hasancebi, S., Tuzkaya, G., & Kilic, H. S. (2023). A Fuzzy Mixed-Integer Linear Programming Model for Aircraft Maintenance Workforce Optimization. In C. Kahraman, I. U. Sari, B. Oztaysi, S. Cebi, S. Cevik Onar, & A. Ç. Tolga (Eds.), *Intelligent and Fuzzy Systems* (Vol. 758, pp. 499–506). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-39774-5_56
- Hashemi, V., Chen, M., & Fang, L. (2016). Modeling and analysis of aerospace remanufacturing systems with scenario analysis. *The International Journal of Advanced Manufacturing Technology*, 87(5–8), 2135–2151. <https://doi.org/10.1007/s00170-016-8566-8>
- Held, M., & Karp, R. M. (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1), 6–25. <https://doi.org/10.1007/BF01584070>
- Hong, I.-H., Chou, C.-C., & Lee, P.-K. (2019). Admission control in queue-time loop production-mixed integer programming with Lagrangian relaxation (MIPLAR). *Computers & Industrial Engineering*, 129, 417–425.
- Hooker, J. N. (2024). *Integer Programming: Lagrangian Relaxation*.
- Hu, S., Dessouky, M. M., Uhan, N. A., & Vayanos, P. (2021). Cost-sharing mechanism design for ride-sharing. *Transportation Research Part B: Methodological*, 150, 410–434.
- Hu, Y., Miao, X., Zhang, J., Liu, J., & Pan, E. (2021). Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization. *Computers & Industrial Engineering*, 153, 107056.
- Huang, R., Riddle, M., Graziano, D., Warren, J., Das, S., Nimbalkar, S., Cresko, J., & Masanet, E. (2016). Energy and emissions saving potential of additive manufacturing: The case of lightweight aircraft components. *Journal of Cleaner Production*, 135, 1559–1570. <https://doi.org/10.1016/j.jclepro.2015.04.109>

- Ichou, S. (2023). Maintenance 4.0: Automation of aircraft maintenance operational processes. *International Journal of Aviation Science and Technology*, 4(01), 23–31.
- IFS. (2019). *IFS Maintenix: Aircraft Fleet Management Software – A New Approach to Maintenance Planning*. <https://www.ifs.com/corp/solutions/ifs-maintenix/fleet-planner/>
- International Air Transport Association. (2017). *IATA Annual Review 2017*. International Air Transport Association.
- International Air Transport Association. (2018). *IATA Annual Review 2018*. International Air Transport Association.
- International Air Transport Association. (2019). *IATA Annual Review 2019*. International Air Transport Association.
- Johnson, P. K., Kulasiri, D., & Sedcole, J. R. (1999). *SIMMAT: An aircraft materials remanufacturing system simulation model. Part I: background and concept*.
- Karaoğlu, U., Mbah, O., & Zeeshan, Q. (2023). Applications of machine learning in aircraft maintenance. *Journal of Engineering Management and Systems Engineering*, 2(1), 76–95.
- Kaskavelis, C. A., & Caramanis, M. C. (1998). Efficient Lagrangian relaxation algorithms for industry size job-shop scheduling problems. *IIE Transactions*, 30(11), 1085–1097.
<https://doi.org/10.1023/A:1007515931780>
- Kenné, J.-P., Dejax, P., & Gharbi, A. (2012). Production planning of a hybrid manufacturing–remanufacturing system under uncertainty within a closed-loop supply chain. *International Journal of Production Economics*, 135(1), 81–93.
<https://doi.org/10.1016/j.ijpe.2010.10.026>

- Kennet, D. M. (1994). A structural model of aircraft engine maintenance. *Journal of Applied Econometrics*, 9(4), 351–368. <https://doi.org/10.1002/jae.3950090405>
- Kinnison, H. A., & Siddiqui, T. (2013). *Aviation Maintenance Management* (2nd ed.). McGraw-Hill Education.
- Kozik. (2016). *The application of ontology in forecasting the demand for spare parts*. <https://doi.org/10.21008/J.2083-4950.2016.6.4.1>
- Kozik, P., & Sep, J. (2012). Aircraft engine overhaul demand forecasting using ANN. *Management and Production Engineering Review*. <https://journals.pan.pl/dlibra/docmetadata?showContent=true&id=89221&language=pl>
- Kulik, A., Shachnai, H., & Tamir, G. (2021). On Lagrangian relaxation for constrained maximization and reoptimization problems. *Discrete Applied Mathematics*, 296, 164–178. <https://doi.org/10.1016/j.dam.2020.10.001>
- Lagos, C., Delgado, F., & Klapp, M. A. (2020). Dynamic Optimization for Airline Maintenance Operations. *Transportation Science*, 54(4), 998–1015. <https://doi.org/10.1287/trsc.2020.0984>
- Lee, Y. W., Allen, T., & Wang, R. Y. (2001). Information Products for Remanufacturing: Tracing the Repair of an Aircraft Fuel-Pump. *IQ*, 77–82. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8a4aad171121ba624cdd658549fe4ae78d6c8333>
- Liu, A., Luh, P. B., Yan, B., & Bragin, M. A. (2021). A novel integer linear programming formulation for job-shop scheduling problems. *IEEE Robotics and Automation Letters*, 6(3), 5937–5944.

- Liu, G., Guo, R., & Chen, J. (2023). A scheduling model of civil aircraft maintenance stand based on spatiotemporal constraints. *Aircraft Engineering and Aerospace Technology*, 95(10), 1518–1530.
- Liu, Y., Wang, T., Zhang, H., Cheutet, V., & Shen, G. (2019). The design and simulation of an autonomous system for aircraft maintenance scheduling. *Computers & Industrial Engineering*, 137, 106041.
- Ma, Y.-S., Jun, H.-B., Kim, H.-W., & Lee, D.-H. (2011). Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. *International Journal of Production Research*, 49(23), 7007–7027.
<https://doi.org/10.1080/00207543.2010.495089>
- Macedo, P. B., Alem, D., Santos, M., Junior, M. L., & Moreno, A. (2016). Hybrid manufacturing and remanufacturing lot-sizing problem with stochastic demand, return, and setup costs. *The International Journal of Advanced Manufacturing Technology*, 82(5–8), 1241–1257.
<https://doi.org/10.1007/s00170-015-7445-z>
- Mariescu-Istodor, R., & Fränti, P. (2021). Solving the Large-Scale TSP Problem in 1 h: Santa Claus Challenge 2020. *Frontiers in Robotics and AI*, 8, 689908.
<https://doi.org/10.3389/frobt.2021.689908>
- Morin, P.-A., Artigues, C., & Haït, A. (2016). *A new relaxation of the Resource-Constrained Project Scheduling Problem*.
- Muckstadt, J. A., & Koenig, S. A. (1977). An Application of Lagrangian Relaxation to Scheduling in Power-Generation Systems. *Operations Research*, 25(3), 387–403.
<https://doi.org/10.1287/opre.25.3.387>

- Pang, X., Gao, L., Pan, Q., Tian, W., & Yu, S. (2017). A novel Lagrangian relaxation level approach for scheduling steelmaking-refining-continuous casting production. *Journal of Central South University*, 24(2), 467–477. <https://doi.org/10.1007/s11771-017-3449-1>
- Pimapunsri, K., & Weerananant, D. (2018). *Solving Complexity and Resource-Constrained Project Scheduling Problem in Aircraft Heavy Maintenance*. 13(11).
- Pimapunsri, K., Weerananant, D., & Riel, A. (2022). *Genetic algorithms for the resource-constrained project scheduling problem in aircraft heavy maintenance* (No. arXiv:2208.07169). arXiv. <https://doi.org/10.48550/arXiv.2208.07169>
- Pogačnik, B., Duhovnik, J., & Tavčar, J. (2017). Aircraft fault forecasting at maintenance service on the basis of historic data and aircraft parameters. *Eksploatacja i Niezawodność - Maintenance and Reliability*, 19(4), 624–633. <https://doi.org/10.17531/ein.2017.4.17>
- Polyak, B. T. (1967). A general method for solving extremal problems. *Soviet Mathematics Doklady*, 8, 593–597. <https://cir.nii.ac.jp/crid/1570291224517467520>
- Qin, Y., Ma, H.-L., Chan, F. T. S., & Khan, W. A. (2020). A scenario-based stochastic programming approach for aircraft expendable and rotatable spare parts planning in MRO provider. *Industrial Management & Data Systems*, 120(9), 1635–1657. <https://doi.org/10.1108/IMDS-03-2020-0131>
- Qin, Y., Wang, Z. X., Chan, F. T. S., Chung, S. H., & Qu, T. (2019). A mathematical model and algorithms for the aircraft hangar maintenance scheduling problem. *Applied Mathematical Modelling*, 67, 491–509. <https://doi.org/10.1016/j.apm.2018.11.008>
- Qin, Y., Zhang, J. H., Chan, F. T., Chung, S. H., Niu, B., & Qu, T. (2020). A two-stage optimization approach for aircraft hangar maintenance planning and staff assignment

- problems under MRO outsourcing mode. *Computers & Industrial Engineering*, 146, 106607.
- Rajamani, R., Wang, J., & Jeong, K.-Y. (2004). Condition-based maintenance for aircraft engines. *Turbo Expo: Power for Land, Sea, and Air*, 41677, 819–823.
<https://asmedigitalcollection.asme.org/GT/proceedings-abstract/GT2004/819/299898>
- Reményi, C., & Staudacher, S. (2014). Systematic simulation based approach for the identification and implementation of a scheduling rule in the aircraft engine maintenance. *International Journal of Production Economics*, 147, 94–107.
- Rivas, A., Delipei, G. K., & Hou, J. (2022). Predictions of component remaining useful lifetime using Bayesian neural network. *Progress in Nuclear Energy*, 146, 104143.
- Samaranayake, P., & Kiridena, S. (2012). Aircraft maintenance planning and scheduling: An integrated framework. *Journal of Quality in Maintenance Engineering*, 18(4), 432–453.
<https://doi.org/10.1108/13552511211281598>
- Samaranayake, P., Lewis, G. S., Woxvold, E. R. A., & Toncich, D. (2002). Development of engineering structures for scheduling and control of aircraft maintenance. *International Journal of Operations & Production Management*, 22(8), 843–867.
<https://doi.org/10.1108/01443570210436172>
- Saraçyakupoğlu, T. (2021, November 6). *The implementation of additive manufacturing technologies in aircraft maintenance, repair, and overhaul (MRO) operations*. 1. Uluslararası Hasankeyf Bilimsel Çalışmalar ve İnovasyon Kongresi, Batman, Turkey.
- Semaan, N. M., & Yehia, N. (2019). A stochastic detailed scheduling model for periodic maintenance of military rotorcraft. *Aircraft Engineering and Aerospace Technology*, 91(9), 1195–1204. <https://doi.org/10.1108/AEAT-09-2018-0254>

- Shirinfar, M., Nasab, H. H., Sadegheih, A., & Fakhrzad, M. (2022). Aircraft Component Spare Parts Logistics and Operation Planning: A Case Study in FARSCO Aircraft MRO Center. *International Journal of Supply & Operations Management*, 9(4).
<https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=23831359&AN=160475844&h=7NJOmrXki8vB84R5J6BUdSqeOKnDhmyu9JhpaOfiS%2F5u63bGdqrprTL49UkIHKbRWcZLxOhTXrsyx%2F3JMUKFUA%3D%3D&crl=c>
- Soleimani, H., Chhetri, P., Fathollahi-Fard, A. M., Mirzapour Al-e-Hashem, S. M. J., & Shahparvari, S. (2022). Sustainable closed-loop supply chain with energy efficiency: Lagrangian relaxation, reformulations and heuristics. *Annals of Operations Research*, 318(1), 531–556. <https://doi.org/10.1007/s10479-022-04661-z>
- Song, X., Cheng, Y., Ren, J., & Tao, F. (2022). A self-learning state-aware rescheduling approach for robot-aided aero-engines fleet inspection. *Journal of Manufacturing Systems*, 65, 802–810.
- Sriram, C., & Haghani, A. (2003). An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*, 37(1), 29–48.
- Stoller, G. (2010, February 3). Planes with maintenance problems have flown anyway. *USA Today*.
- Stranjak, A., Dutta, P. S., Ebden, M., Vytelingum, P., & Rogers, A. (2008). *A multi-agent simulation system for prediction and scheduling of aero engine overhaul*.
<https://eprints.soton.ac.uk/265651>
- Sun, L., Yang, R., Feng, J., & Guo, G. (2024). Applications of Lagrangian relaxation-based algorithms to industrial scheduling problems, especially in production workshop

- scenarios: A review. *Journal of Process Control*, 139, 103233.
<https://doi.org/10.1016/j.jprocont.2024.103233>
- Table, A. (2025). A lagrange relaxation based algorithm for parallel injection machine scheduling problem. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 40(1), 277–286.
- Tolouei, K., Moosavi ,Ehsan, Bangian Tabrizi ,Amir Hossein, & and Afzal, P. (2021). Application of an improved Lagrangian relaxation approach in the constrained long-term production scheduling problem under grade uncertainty. *Engineering Optimization*, 53(5), 735–753. <https://doi.org/10.1080/0305215X.2020.1746295>
- Torkashvanda, M., Shamami, N., & Bigdelic, H. (2023). *The Equipment Scheduling and Assignment Problem in the Overhaul Industry*.
https://www.academia.edu/download/108570706/article_169091_d4e11a805c1c9293f326229c3a256e6c.pdf
- Tretheway, M. W., & Markhvida, K. (2013). *Airports in the aviation value chain: Financing, returns, risk and investment* [Discussion Paper]. International Transport Forum.
- Tseremoglou, I., Van Kessel, P. J., & Santos, B. F. (2023). A Comparative Study of Optimization Models for Condition-Based Maintenance Scheduling of an Aircraft Fleet. *Aerospace*, 10(2), 120. <https://doi.org/10.3390/aerospace10020120>
- U.S. Bureau of Labor Statistics. (n.d.). *Occupational Employment and Wage Statistics (OEWS) Tables*. U.S. Department of Labor. Retrieved June 30, 2025, from <https://www.bls.gov/oes/current/oes493011.htm>
- Van Loon, P., & Van Wassenhove, L. N. (2018). Assessing the economic and environmental impact of remanufacturing: A decision support tool for OEM suppliers. *International*

- Journal of Production Research*, 56(4), 1662–1674.
<https://doi.org/10.1080/00207543.2017.1367107>
- Vercraene, S., Gayon, J.-P., & Flapper, S. D. (2014). Coordination of manufacturing, remanufacturing and returns acceptance in hybrid manufacturing/remanufacturing systems. *International Journal of Production Economics*, 148, 62–70.
- Verhagen, W. J. C., & De Boer, L. W. M. (2018). Predictive maintenance for aircraft components using proportional hazard models. *Journal of Industrial Information Integration*, 12, 23–30. <https://doi.org/10.1016/j.jii.2018.04.004>
- Wakiru, J., Pintelon, L., Muchiri, P. N., & Chemweno, P. (2018). Maintenance optimization: Application of remanufacturing and repair strategies. *Procedia CIRP*, 69, 899–904.
- Wang, L., Chen, Y., Zhao, X., & Xiang, J. (2024). Predictive maintenance scheduling for aircraft engines based on remaining useful life prediction. *IEEE Internet of Things Journal*.
<https://ieeexplore.ieee.org/abstract/document/10472322/>
- Wang, Y., Jiang, Z., Hu, X., & Li, C. (2020). Optimization of reconditioning scheme for remanufacturing of used parts based on failure characteristics. *Robotics and Computer-Integrated Manufacturing*, 61, 101833.
- Weide, T. V. D., Deng, Q., & Santos, B. F. (2022). Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers & Operations Research*, 141, 105667. <https://doi.org/10.1016/j.cor.2021.105667>
- Yan, S., Chen, C. Y., & Yuan, C. Y. (2008). Long-term aircraft maintenance scheduling for an aircraft maintenance centre: A case study. *International Journal of Applied Management Science*, 1(2), 143. <https://doi.org/10.1504/IJAMS.2008.021098>

- Yang, Q., & Zheng, L. (2016). Managing coordination complexity in the remanufacturing of aircraft engines. *Frontiers of Engineering Management*, 3(2), 158–164.
- Zhang, J., Liu, X., & Tu, Y. L. (2011). A capacitated production planning problem for closed-loop supply chain with remanufacturing. *The International Journal of Advanced Manufacturing Technology*, 54(5–8), 757–766. <https://doi.org/10.1007/s00170-010-2948-0>
- Zhao, X., Luh, P. B., & Wang, J. (1999). Surrogate Gradient Algorithm for Lagrangian Relaxation. *Journal of Optimization Theory and Applications*, 100(3), 699–712. <https://doi.org/10.1023/A:1022646725208>
- Zheng, P., Wang, J., Zhang, J., Yang, C., & Jin, Y. (2019). An adaptive CGAN/IRF-based rescheduling strategy for aircraft parts remanufacturing system under dynamic environment. *Robotics and Computer-Integrated Manufacturing*, 58, 230–238. <https://doi.org/10.1016/j.rcim.2019.02.008>