Aircraft Maintenance, Repair & Overhaul Spare Parts Management

Demand & Procurement Optimization

Christos Paschalidis

TU Delft x nlr

MSc thesis in TIL

Aircraft Maintenance, Repair & Overhaul Spare Parts Management

Demand & Procurement Optimization

Christos Paschalidis

November 2025

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Transport, Infrastructure & Logistics

Christos Paschalidis: *Aircraft Maintenance, Repair & Overhaul Spare Parts Management* (2025)

The work in this thesis was carried out in collaboration with:

Royal Netherlands Aerospace Centre

Supervisors: Dr. F. Schulte
Dr. M.J. Ribeiro
Co-reader: Dr. I.I. de Pater

Report ID: 2025.TIL.9139

Abstract

Managing spare parts for Aircraft Maintenance, Repair, and Overhaul (MRO) is challenging because there is a significant gap between long-term maintenance schedules and daily procurement decisions. While existing research often addresses demand forecasting and inventory control in isolation using abstract assumptions, a new framework is presented to bridge this gap by directly connecting day-to-day procurement decisions with the fixed, fleet-wide maintenance schedule. A task-based approach enhances traditional planning, which often relies on aggregate forecasts and can miss the specific needs of individual checks. The result is a transparent, cost-based model built for operational utility, where every decision accounts for probabilistic predictions and remains auditable. This traceability is crucial in an environment where complete historical data is often unavailable.

The framework consists of two modular stages. First, a demand forecasting methodology converts raw maintenance tasks into a usable, time-phased, and probabilistic demand signal. To accomplish this, maintenance tasks are systematically grouped based on their technical attributes. The outcome is a repeatable method to describe the demand potential of each scheduled task.

Second, a daily procurement optimization model was created to act on this detailed forecast. The algorithm replicates a planner's decision-making process by explicitly comparing the expected future costs of buying, waiting, or selling surplus stock. To mirror operational reality, the model utilizes regular orders with uncertain lead times, reactive express orders, and pre-procurement. Every decision becomes a justifiable trade-off regarding the cost of purchasing and holding inventory, and the high financial penalty of a stockout.

Finally, the model was validated against two benchmarks: a fully conservative (100% service) strategy and a standard periodic-review policy. The proposed model reduced total net costs for both benchmarks, achieving savings of 17.5% and 9.2% respectively. The analysis shows this advantage stems from the strategic acceptance of controlled risks when doing so leads to a lower expected total cost. Further sensitivity analysis revealed that the cost-driven logic is robust even under poor forecasts, as it automatically compensates to maintain a safe inventory level. The analysis also identifies key non-linear trade-offs, finding that total net cost is minimized at a moderate level of caution regarding stockout penalties, rather than at the extremes of under- or over-estimation. The framework ultimately provides a practical and transparent decision support tool, demonstrating that a task-specific, dynamic, and cost-based approach is more effective and resilient than traditional, static planning rules.

Acknowledgements

A special thanks must first go to the group of people who had absolutely no obligation to assist me. Despite receiving an unsolicited email from a student in another country, they generously decided to allocate their valuable time to share their knowledge. Specifically, I want to thank Jimmy and Jonathan from ST Engineering, who kindheartedly shared a wealth of information about how the inventory management industry works (and, more importantly, how it doesn't). Thank you to Rene from KLM, who explained in detail how complex issues are actually handled in practice, and finally, to my compatriot and veteran in inventory forecasting, Professor Aris Syntetos.

I want to thank them because they knew they weren't getting anything from me in return, yet they still offered their help. As I promised each of them, I will do my best to pay it forward when and if a student comes to me with a similar request in the future.

To my friends and fellow students, thank you for the shared misery in the university library. Those tough, quiet summer days in an empty building would have been unbearable without you. And let's be honest, sometimes it's just nice to see people who are even more stressed than you are.

To my wonderful family, who have absolutely no idea what I am studying, let alone what this thesis is about. Thank you for your firm moral and financial (mostly the second) support. I know you've all just been patiently waiting for me to finally graduate and become unemployed so I would return home to you.

And finally, I must thank the most supportive supervision team a student could ask for: dr. Frederik Schulte, dr. Marta Ribeiro, and Arjan Derks. Thank you for bearing with me for seven months and putting up with my unconventional methods. I am incredibly grateful that you gave me the freedom to explore this problem in the way that I wanted to, and for gently intervening only when I was getting completely off track.

"Everything will go according to plan,
unless you expect it to."

– Christos Paschalidis

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

The efficiency of aircraft maintenance is built on a detailed, long-term schedule. This thesis presents a framework that uses this schedule as a direct input for spare parts procurement, creating a powerful link between the long-range plan and daily operational decisions. It contributes a focused, usable bridge with two parts: (i) a forecasting layer that assigns a replacement probability to each planned task, and (ii) an optimization layer that turns those probabilities into calendar-dated order decisions. The aim is decision support that planners can audit, showing not only what to buy, but why that choice is the most sensible.

A practical tool must align with how planners work, how risk is managed, and how purchasing actually happens. This thesis therefore follows three core design principles. First, calendar fidelity: decisions must be made on the same daily timeline as the maintenance plan, not in abstract periods. Second, cost traceability: every recommendation must clearly expose the few key drivers behind it, such as replacement likelihood, on-hand stock, and lead time risk, in a way a planner can check. Third, operational fit: the outputs must be immediately usable as purchase orders without requiring new or complex workflows. These principles keep the focus on usefulness; a solution is only good if it can be explained quickly and executed easily.

1.1 Background and Societal Context

Effective spare parts provisioning is a central enabler of scheduled aircraft maintenance. Each maintenance task needs its specific parts on hand to begin and finish on time. Airlines therefore aim for the maximum possible service level at the minimum total cost. Service quality links directly to safety and reliability targets because the timely replacement of worn parts keeps failure risk low. A sound provisioning policy also has financial value by reducing unplanned purchases, avoiding premium freight, and limiting capital tied up in excess stock. Environmental goals enter the picture as well because extending component life and cutting scrap volumes reduces material waste. Finally, good stock availability lets licensed technicians use their shift hours productively, which protects an airline's investment in skilled labor.

Meeting these objectives is difficult in practice. A modern narrow body fleet contains more than one million distinct part numbers, with several units of many items installed in each aircraft. Parts differ widely in cost, lead time, and expected failure probability. Some items appear on the Minimum Equipment List, so their absence can prevent a flight from departing. Other items affect passenger comfort but not safety, and they still carry commercial importance. Inventory planners must balance cash tied up in stock against the cost of disrupted operations if parts are unavailable. They also face constraints such as limited storage space, supplier capacity, and budget ceilings. The practical question is how to develop data driven rules that translate the maintenance plan into timely purchase orders. These rules need to secure service targets while holding inventory cost and operational impact to a minimum.

The spare parts classification scheme explained in this section follows the structure given by International Air Transport Association [2015]. That document offers a shared language, meant as a guide, that links engineering, supply, and finance disciplines, helping them apply suitable stock levels, repair arrangements, and response times. Airlines and maintenance providers adapt the labels and thresholds to suit fleet mix, contract terms, and information systems.

IATA separates parts by three independent criteria. **Scrap rate** indicates how often a component is discarded rather than repaired. **Financial** treatment shows whether the item remains on the balance sheet as a depreciating asset or is expensed when it is issued for use. **Life-cycle profile** records whether the unit can circulate indefinitely, only for a limited number of repairs, or for a single service event. These three criteria provide a structured way to connect technical characteristics of spare parts to how they are handled in terms of cost, stock levels, and repair policies.

Combining the three criteria produces three widely used classes. **Rotable** inventory covers high value assemblies such as wheels, brakes, fuel pumps, or radar transceivers. These parts have a very low scrap rate, can be overhauled many times, and are depreciated as fixed assets. **Repairable** inventory contains medium value units like oxygen bottles, starter motors, or certain lights. They can be restored but show significant scrap rates, so fresh units are needed after several repair loops. These items appear as inventory assets and demand close tracking of repair turnaround and replenishment lead time. **Expendable** inventory includes single use parts such as filters, seals, fasteners, and gaskets. They are consumed once and expensed at issue. The three way split supports practical policy choices. Rotables may justify dedicated safety stock or access to pooling programmes to avoid operational disruptions, while still keeping inventory costs under control. Repairables benefit from scrap aware sourcing and disciplined lead time control. Expendables are well suited to economic order quantity rules, vendor managed programs, and lot based traceability. Regardless of how each operator fine-tunes the details, this simple classification remains sufficient to handle spare parts effectively.

In most airline organisations, the responsibility for spare parts planning is handled by teams that work across engineering, maintenance, and supply chain functions. The way provisioning is approached depends heavily on the airline's broader business setup. Factors like network structure, fleet choice, dispatch reliability targets, and budget all influence how spare parts decisions are made. Once these general conditions are clear, planners typically work through a structured process to decide which parts should be stocked, where to keep them, and how to manage ownership and repair. Industry material, like the IATA guidelines, proposes a sequence for these decisions that helps avoid early mistakes causing unnecessary costs or shortages later on. While different airlines may apply this process in their own way, the overall logic of moving from business objectives to technical and operational decisions remains similar.

Once the basic demand and network picture are understood, the more practical side of provisioning follows. A key question is where to keep inventory and in what quantities. For example, airlines operating with a hub-and-spoke network often place critical parts at major hubs, while relying on pooling or borrowing at smaller outstations. Airlines with more point-to-point or charter operations may lean on portable kits or external suppliers to avoid holding large amounts of stock. Another decision is how the airline chooses to own, lease, or outsource parts, which depends on factors like part cost, lead time, and how critical the part is to keeping aircraft serviceable. In most cases, a mix of different options is used. Airlines also keep contingency measures in place, such as short-term borrowing or emergency purchasing, to deal with unexpected failures or supply delays. In the end, how an airline manages provisioning depends on both its technical needs and its business model.

1.2 Scope, Methodology & Practical Relevance

This thesis positions itself at the interface between an existing maintenance planning tool and the spare parts procurement process that follows. The model respects the permissible start–finish windows defined by the planning system. Once a task is scheduled within its window, it is treated as fixed for the purposes of materials planning; provisioning decisions do not move tasks. Work confined to unscheduled events, ad-hoc line maintenance, or real-time slot negotiation lies outside the study. The physical setting is limited to a single, central inventory location, and transport lead times are handled as aggregate parameters. Financial ceilings, dispatch targets, and approval workflows are treated as fixed constraints. By limiting the study to scheduled checks and a single stock point, the analysis concentrates on translating the dated maintenance workload generated by the planning tool into concrete, part-specific procurement requirements.

The conceptual stance is to build a new but literature-supported procedure that links each maintenance task to a component, a replacement probability, and its timing. Existing work on demand forecasting, parts classification, and aviation logistics supplies many of the ideas, yet the procedure is assembled from scratch to suit the data and constraints described above. In the absence of historical replacement records, the framework uses synthetic reliability inputs, clearly flagged and quantitatively described so they can be replaced as soon as actual observations become available. Every assumption is documented to keep the process transparent and to let future users update values without rewriting core steps. The intended result is a repeatable path that starts with a time-stamped maintenance plan and ends with an itemized purchase list ready for the existing procurement system, while remaining flexible enough to accept richer data or broader objectives in the future.

The study does not optimize or reschedule the maintenance plan. Inputs are the dated tasks supplied by the planning system. This focus isolates the core question of how to time and size material orders once the plan is known. The analysis does not cover unplanned maintenance or aircraft on ground events. Those cases follow a different control loop with very short lead times. Network placement, multi echelon stocking, and moves between stores are also not modelled. The work uses one central stock point so that results are not mixed with allocation effects across many sites.

The model does not select vendors or contracts and does not plan repair shop capacity. These enter the analysis through parameters such as lead time distributions, repair turnaround, and price. Ownership choices such as lease, buy, and pool are taken as given and are reflected in cost and availability. Serial number rules, lot traceability, and deep engineering detail on specific components are not expanded, both because data is often proprietary and because these details are not essential to the timing problem that is studied here. These limits keep the focus on the main contribution, which is a clear and auditable link from a dated plan to dated purchase decisions, while leaving a clean path for future extensions.

Developing a parts provisioning tool requires a systematic approach to connect the maintenance schedule with spare part demand. This entails creating a clear methodology that utilizes all available information about each maintenance task to forecast the required spare parts. The objective is to establish a structured process that translates task details into actionable demand data, where each dated task is associated with its target component and a quantitative estimate of replacement need on that date. Replacements that are certain are assigned a probability of 1, while those that are uncertain are assigned a probability based on task characteristics and available condition cues, resulting in a calendar-aligned demand file that includes task, component, date, expected quantity, and supporting assumptions, allowing for auditing and updating of values as needed.

The decision procedure is the crucial step, generating actionable parts provisioning decisions based on key inputs. It operates on the same calendar and takes as inputs on hand balances, open orders, lead time distributions, service targets, and budget limits. The procedure defines the action space over timing and quantity, including explicit criteria for when an express action is warranted. Recommendations must be traceable to a small set of drivers, robust to modest input changes, and fast to recompute when the plan updates. Compatibility with the current procurement process is mandatory, and the procedure does not create new workflows.

Airlines invest heavily in modern planning tools, yet spare parts decisions are still often driven by ad-hoc rules and urgent phone calls. Late deliveries can push a hangar visit beyond its slot, disrupt the flying programme, and trigger premium freight. Stocking large buffers prevents those shocks but ties up capital in slow-moving material. The procedure developed here offers a middle path. By converting the dated list of maintenance tasks into an itemized, time-phased view of expected demand, it supports on-time targeted purchases. Finance gains by releasing cash otherwise frozen in surplus stock, maintenance control sees fewer last-minute searches, and the workload on veteran planners is eased because key assumptions are recorded rather than held only in personal experience.

Progress in spare parts planning delivers a second benefit by supporting an Equalized Maintenance Program. An EMP spreads heavy work across the year rather than clustering it in a few large checks, keeping aircraft on the line for longer stretches and smoothing labour demand. For such a program to work, every supporting activity must be controlled, from manpower planning and tooling to the flow of parts. If material supply is uncertain, the entire equalized plan can slip and generate unplanned downtime. A spare parts system that records lead times, stock levels, and usage in real time removes that weak link. The framework outlined here provides that capability, giving planners a transparent view of future demand and allowing the EMP to achieve both lower ground time and more even resource use.

Taken together, the elements form a decision support framework with calendar dated states and an explicit set of possible actions. On each date the state is defined by on hand stock, open orders, expected demand from scheduled tasks, and budget. For each component the tool evaluates actions such as buy now, buy later, buy express, or hold, and for each action it estimates the expected effect on service risk, holding cost, and budget use. The recommended action is shown with its value, the gap to the next best choice, and the small set of drivers that explain the result. Planners see not only what to do but also why that choice dominates and what change in inputs would switch the decision. Recommendations are presented in a format that allows quick validation, optional overrides, and direct entry into the existing purchasing workflow.

1.3 Research Question(s)

To direct the research, all endeavours are based on the following research question:

> *RQ: How can a maintenance plan be translated into a procurement strategy that minimizes cost by leveraging task-specific demand forecasts?*

To structurally answer the research question, a set of sub-questions are investigated and answered.

- SQ1: How can each scheduled maintenance task be mapped to its exact spare part requirement?
- SQ2:Which uncertainty drivers are most relevant and how can they be represented in a model?
- SQ3: How can the resulting procurement problem be formulated and solved as a tractable optimization model?
- SQ4a: Which performance indicators best capture cost and service trade-offs, and how can their impact be tested through sensitivity analysis?
- SQ4b: How does permitting in-horizon sales of surplus parts influence procurement choices and total cost?

1.4 Thesis Outline

The research begins by reviewing existing literature on spare parts classification, demand forecasting for intermittent demand, and integrated inventory and maintenance planning (Chapter 2). Following this, Chapter 3 details the first major part of the framework: a complete methodology for demand forecasting. It explains how the raw Maintenance Planning Document (MPD) is transformed into a time-phased, probabilistic demand signal for individual parts. This section covers the use of Natural Language Processing (NLP) to enrich task descriptions, the clustering of tasks based on their technical attributes, and the application of reliability models to generate a five-year simulated demand forecast.

Chapter 4 introduces the core procurement and inventory optimization model. It defines the daily, cost-based logic that allows the model to transparently weigh the costs of buying, waiting, or selling surplus stock. The model's performance is then validated by benchmarking it against both a fully conservative strategy and a 12-month periodic review policy. Chapter 5 demonstrates the framework's use as a decision support tool, presenting a sensitivity analysis on key drivers like supplier lead time, demand accuracy, and penalty costs. Finally, the thesis concludes by summarizing the findings (Chapter 6) and discussing the framework's limitations and practical extensions (Chapter 7).

2 Related work

This section reviews key areas of research on spare parts management for aircraft MRO. First, standalone techniques that treat demand independently of the maintenance calendar are examined, including part classification, time series forecasting for intermittent demand, and inventory policies under uncertainty. Then, calendar aligned methods that derive demand from maintenance schedules and inspection outcomes are covered, showing how plan driven demand can be modeled. Next, integrated frameworks that couple maintenance planning with procurement decisions are explored. The section ends with a brief summary of gaps in current work and a note on how this thesis contributes to the literature.

2.1 Spare Parts Classification

Traditional single-factor classifications (e.g. ABC analysis by annual usage value) often fail to capture the diverse drivers of spare parts management [Roda et al., 2014]. Qualitative schemes such as VED depend on maintenance engineers or logistics staff assigning parts to “Vital”, “Essential”, or “Desirable” groups by experience rather than measurable rules; classifications can therefore vary between assessors, drift over time as personnel or fleet priorities change, and lack reproducible weighting of cost, risk, and service consequences. Current research therefore emphasizes multi-criteria frameworks that consider factors such as part criticality, demand frequency, lead time, and cost simultaneously [Roda et al., 2014].

The review by Bacchetti and Saccani [2012] revealed that, while literature proposes advanced multi-criteria classifications, many companies still use simplistic methods in practice. They highlighted a need to bridge this gap by developing practical classification tools that practitioners can adopt. Recent studies have begun to address this. For instance, Ayu Nariswari et al. [2019] developed an Analytical Hierarchy Process (AHP) based model for an Indonesian airline MRO. Their multi-criteria approach (incorporating usage rate, criticality, lead time, etc.) produced a more accurate and transparent spare parts prioritization than the MRO’s existing ABC-like system.

2.2 Demand Generation Modelling

For tasks that mandate automatic replacement at fixed intervals, demand is treated as fully deterministic and tied to the overhaul dates specified in the maintenance plan. When replacement is conditional on inspection results, each scheduled inspection can be modelled as a Bernoulli trial. Zhu et al. [2020] combine the MPD’s inspection counts with task-specific replacement probabilities and obtain a binomial (or Poisson-binomial) distribution for short-term demand. Delay-time models account for how defects develop between inspections. In Wang and Syntetos [2011], the probability that a part is replaced at the next scheduled check is calculated directly from the inspection interval and the chosen delay-time distribution. Installed-base approaches replace the event count by information on part condition. Deshpande et al. [2006] use the age profile of parts in service to estimate near-term replacements, while Hu et al. [2015] generalise this to a two-trigger rule where either a calendar limit or a usage threshold prompts replacement. A related two-step framework by Romeijnders et al. [2012] treats the number of repairs scheduled in the maintenance plan as known and multiplies it by an average parts-per-repair factor to forecast total demand.

For corrective or unscheduled demand, authors have largely adopted stochastic frameworks that treat part requests as intermittent events rather than continuous flows. Early work by Croston [1972] introduced a procedure that separates the estimation of demand sizes from the intervals between occurrences, a structure suited to low-frequency usage. Subsequent refinements, notably the Syntetos–Boylan adjustment [Syntetos and Boylan, 2005], retain Croston's separation principle while correcting its tendency to over-forecast, and have become a common baseline for intermittent-demand studies.

One complementary approach to spare parts demand forecasting leverages reliability engineering and installed-base information. Instead of relying purely on time-series patterns, this method links demand directly to equipment failure processes and the population of units in service. Van der Auweraer and Boute [2019] develop a model that predicts spare part demand by tracking the active installed base of machines and each part's failure distribution. The idea is to estimate when components will fail (using reliability curves or failure rates) given the number of machines in use and their maintenance schedules, thereby forecasting a probability distribution of spare part demand over the lead time. Reliability-based forecasts link the active installed base and component failure curves to predict future part demand, then set inventory parameters such as base-stock levels. This study shows that these failure-driven models meet service targets with less stock than traditional intermittent-demand methods, a gain that grows with larger fleets and longer lead times.

2.3 Spare Parts Management under Uncertainty

Classical base stock rules are built around a single demand distribution, usually Poisson, yet demand for spares often drifts or spikes in ways that defy that assumption. Kang et al. [2023] frame inventory planning as an adaptive robust optimisation problem in which demand can follow any path within a predefined uncertainty set; the policy is sized to maintain service targets even under the most adverse path. They solve the model with a column and constraint generation routine and report, in both simulation and an ASML case study, that the robust policy matches fill rates while tying up much less capital than a Poisson based base stock benchmark.

One straightforward tactic is to set fixed thresholds on key inventory signals. Hekimoğlu et al. [2022] present a monitoring scheme for repairable parts that tracks on-hand stock, items in repair and scheduled maintenance tasks; when the queue of parts still in repair passes a critical threshold, selected units are routed through an expedited repair channel at higher cost. The resulting threshold policy cushions downtime more economically than large safety stocks. Related work uses scenario based simulations and chance constrained formulations to set safety stock so that the probability of a stockout stays below a chosen level while accounting for variability in demand and repair turnaround.

Another key perspective is the network-wide optimization of spare stocks across multiple sites, explicitly accounting for uncertainty in demand and supply lead times. In aircraft MRO, airlines often operate distributed inventories (central warehouses and local base stocks), so multi-echelon models have been developed to decide where and how much to stock under stochastic demand [Li et al., 2023]. Further work extends classical METRIC-based approaches by adding practical pooling mechanisms like lateral transshipments, that is, reassigning parts between bases to cover unexpected shortages [Liu et al., 2018].

2.4 Integrated Spare Parts and Maintenance Planning

Rather than treating maintenance scheduling and spare provisioning as separate problems, recent research optimizes them together for better overall performance. Qin et al. [2020] present a scenario-based stochastic programming model that integrates aircraft maintenance plans with spare parts inventory decisions. In their formulation, an MRO provider decides upfront which rotables to overhaul in regular vs. expedited mode, how to rotate serviceable rotables between incoming aircraft, and how many spares to pre-purchase, all before knowing exact future failures. Then, in a second stage, uncertain demand scenarios (unscheduled failures of various parts) are realized, and the model decides on additional emergency procurement to fulfill any shortfalls. This two-stage approach minimizes total cost, including overhaul costs (trading off regular vs. faster turnaround), initial stocking costs, and scenario-weighted penalties for late or excess parts. The authors report significant cost savings and service level improvements compared to treating procurement and maintenance planning independently.

Erkoc and Ertogral [2016] tackle another integrated problem common in aviation MRO: scheduling mandatory component overhauls in a way that respects limited repair-shop capacity and available spares. They developed an integer programming model for an exchange program where rotable parts are swapped out of aircraft at due time and sent for overhaul. Because both the number of spare rotables and the repair throughput are constrained, some overhauls may need to be done early (before their deadline) to avoid bottlenecks. Early removals, however, shorten the useful life of components and are undesirable for airlines. The model therefore optimizes overhaul start times on parallel repair lines and the exchange schedule of rotable units to minimize total earliness (i.e. avoid premature replacements). They prove the linear relaxation of this schedule-inventory model is integer-optimal, enabling a fast exact algorithm.

Additional efficiency can be gained by integrating spare parts planning across multiple assets or even organizations. Research has explored strategies like inventory pooling in aviation supply chains, wherein airlines or regional fleets share critical spares. Integrated models coordinate repair capacity investment with inventory control. Buyukkaramikli et al. [2015] combined decisions on how many spare rotables to stock with how much repair capacity to allocate, finding the optimal balance between holding more spares versus expanding shop throughput. Overall, the literature indicates that tight integration between maintenance planning and spare parts management yields considerable benefits: joint models tend to recommend lower inventory levels than separate models for the same service targets, because they can schedule maintenance dynamically (e.g. slight timing adjustments or resource reallocation) to mitigate parts shortages.

2.5 Synopsis

To consolidate the literature reviewed, Table 2.1 positions key studies against the specific methodological pipeline developed in this thesis. This visual summary helps to map existing research to the sequential stages of the problem, from initial part classification to a final procurement action. The table provides a high-level overview of how each piece of literature contributes to one or more stages of this end-to-end process.

The table shows the evaluation of each study against five key methodological components. Technical Spare Parts Classification refers to methods that group parts based on their technical attributes and operational environment, not just on economic value. Reliability Modelling marks studies that use a formal statistical model to estimate a component's replacement probability as a function of time or use. Plan-based demand forecasting indicates that the model uses dated, specific maintenance tasks from an operational plan as a direct input for forecasting. Calendar aligned procurement decisions identifies policies that generate time-stamped order decisions (e.g., daily) rather than abstract or aggregated reorder levels. Finally, Stochastic Uncertainty Modelling signifies the use of explicit probability distributions to manage uncertainty, as opposed to relying on a few deterministic scenarios.

Table 2.1: Synopsis of Literature Along the Methodological Pipeline

Study	**Technical Spare Parts Classification**	**Reliability Modelling**	**Plan-based Demand Forecasting**	**Calendar Aligned Procurement Decision**	**Stochastic Uncertainty Modelling**	**Notes**
Roda et al. [2014]	✓	–	–	–	–	
Ayu Nariswari et al. [2019]	✓	–	–	–	–	
Zhu et al. [2020]	–	–	✓	*P**	✓	*Monthly aggregation; not daily.
Wang and Syntetos [2011]	–	✓	✓	–	–	
Romeijnders et al. [2012]	–	–	✓	–	–	
Van der Auweraer and Boute [2019]	–	✓	–	–	✓	
Kang et al. [2023]	–	–	–	–	*P**	*Uncertainty sets; no distributional assumptions.
Hekimoğlu et al. [2022]	–	–	–	–	✓	
Qin et al. [2020]	–	–	✓	*P**	✓	*Two-stage (not daily rolling).
Erkoc and Ertogral [2016]	–	–	✓	✓	–	
Buyukkaramikli et al. [2015]	–	–	–	–	✓	
This thesis	✓	✓	✓	✓	✓	

Symbols: ✓ = yes, *P* = partially, – = not addressed.

The classifications in Table 2.1 reflect whether a study contributes a distinct methodological solution for each attribute, rather than merely acknowledging its relevance. Consequently, a checkmark implies an explicit modeling contribution. Furthermore, this framework introduces a decision variable rarely implemented in standard procurement models: the ability to actively sell surplus stock. Integrating this sales mechanism as a dynamic, calendar-aligned operational choice represents a novel addition, particularly within the aviation industry where such bi-directional flows are seldom modeled.

2.6 Conclusion

The literature gives several strong tools for aviation spare parts planning. Studies show how to rank parts with multiple criteria, add inspection and reliability data to demand forecasts, use stochastic or robust models to deal with uncertainty, and link maintenance plans with inventory decisions. Yet nearly all of this work focuses on optimizing static inventory policies, such as safety stock levels, rather than modeling the daily, calendar-based decisions a planner must make. These factors are usually patched in after optimization rather than built into the model itself. Most papers also stop at reporting overall cost or fill-rate improvements and do not produce the dated purchase orders that buyers need. Metrics critical to airline operations, such as capital tied up in inventory or the probability of a no-fill event per flight, are rarely integrated into objective functions. Existing approaches still fall short of explicitly associating real-world maintenance tasks with their specific demand probabilities. Consequently, they lack the granular link needed to convert real-world uncertainties and budget constraints into calendar-dated order plans that are steered by industry KPIs and costs.

2.7 Discussion

In practice, a high-level inventory policy does not tell a planner what to order today. This gap between a long-term rule and a daily decision often means that procurement still relies on experience or simple reorder points. These traditional methods are not easily auditable and may not align with the airline's real financial drivers or risk policies.

A planner must balance the high, immediate cost of a stockout against the slow, long-term cost of holding capital in surplus stock. These are practical, daily, cost-based trade-offs that many published models do not capture . Furthermore, these models often rely on detailed failure histories or condition data that many operators, especially those with new fleets, simply do not collect. Planners need tools that work with the data they do have, such as the maintenance plan itself. This study will aim to address these gaps by integrating practical performance metrics and realistic data constraints directly into the optimization process.

2.7.1 Contribution to Literature

The thesis links topics that are usually treated separately. It keeps the maintenance schedule as the backbone and connects task dated demand construction with procurement timing in one continuous pipeline. Much of the literature studies a single part of this flow, such as text classification of work cards, demand forecasting, or reorder rules, and it often aggregates time into periods that ignore actual task dates. The contribution is a calendar aligned, task level view that lets the interactions between demand modeling, lead times, and purchasing be examined within one consistent setup. This contrasts with period based control and component specific studies that do not test how parts of the process affect one another. The focus is on integration and traceability across the whole flow, rather than on proposing a new forecast model or a new policy in isolation.

Methodologically, this work builds directly upon the stochastic procurement framework of Qin et al. [2020] and the plan-based demand modeling of Zhu et al. [2020]. However, it extends these foundational approaches by introducing a systematic framework for actually calculating the required task-specific failure probabilities, rather than assuming them as exogenous inputs or deriving them solely from aggregate history. By using technical clustering and reliability modelling to generate these probabilities from the maintenance plan itself, the thesis creates a direct, auditable link between the engineering characteristics of a task and the resulting procurement decision. Furthermore, the model expands the standard decision space by introducing a dynamic sales mechanism, a feature rarely modelled in aviation literature that allows the system to bi-directionally optimize inventory levels.

The thesis also contributes through transparency and modularity. Inputs and assumptions are explicit and easy to replace, which is uncommon in data rich or black box approaches. Synthetic parameters can be swapped for operational data without changing the overall flow. Modules for text extraction, clustering, reliability, lead times, and economic settings are exchangeable and documented, which supports replication and transfer to other operators. The emphasis is on a clear architecture that others can adapt to their context.

3 Demand Forecasting

Introduction

Effective spare parts procurement and inventory optimization are fundamentally dependent on a reliable demand forecast. This forecast acts as the primary signal that drives all subsequent decisions regarding stock levels, safety stock calculations, and purchasing schedules. In an ideal operational environment, this forecast would be derived from years of historical maintenance records, capturing the real-world failure rates and replacement patterns of components across the fleet. However, for the scope of this thesis, access to such proprietary historical data is not available. This constraint mirrors real-world scenarios where data is limited, such as for new airline operators or during the introduction of a new fleet type. In these contexts, a prerequisite step is to first establish a logical framework for generating a synthetic demand signal. This chapter is dedicated to developing and detailing such a framework.

The goal is to construct a transparent, adaptable, and defensible methodology for simulating spare part demand. This objective does not involve claiming a perfect prediction of future needs, which would be impossible even with empirical data. While this thesis applies the framework to the Airbus A320 family as a case study, the underlying principles are designed to be transferable to other modern aircraft platforms. The framework is deliberately designed to be modular. For the purposes of this study, the model is populated with a set of reasoned assumptions, yet each parameter, including failure characteristics and component attributes, can be easily adjusted or replaced. This ensures that the model can be tuned by experienced maintenance engineers or refined with actual operational data as it becomes available, enhancing its practical utility beyond a purely theoretical exercise.

This process has several steps to turn the general tasks in the MPD into a detailed demand forecast over time. First, the process uses Natural Language Processing (NLP) to automatically extract key technical details from the task descriptions in the MPD. These new details, along with existing MPD data, are then used to sort the maintenance tasks into logical groups. The goal of this clustering is to group tasks that are expected to have similar failure patterns. This enables the application of a single reliability model to all tasks within a group, simplifying the analysis by leveraging the assumed homogeneity within each group.

For each group of tasks, a statistical model is fitted to predict the chance that doing the task will lead to a part replacement. The predicted failure rates are then utilized to simulate the outcomes of maintenance activities, allowing for the estimation of spare part demand under various scenarios. Specifically, the model is applied to a representative maintenance schedule, generating a forecast that encompasses the expected results of numerous maintenance checks over a specified planning horizon. The final output is a complete forecast of the demand for spare parts.

The next sections of this chapter explain each step of the framework. Beginning with an ”overview” that illustrates the complete logical sequence, from the initial MPD data to the final demand forecast, and briefly introduces the tools used. Each of these stages is then explored in greater detail in their dedicated sections, covering the NLP enrichment, task clustering, reliability modeling, and the simulation of operations. The chapter concludes by presenting the resulting spare part demand forecast, which provides the foundation for the inventory strategies explored later in this thesis.

Overview

The first step in the process handles the unstructured text within the MPD's "Description" field (see Appendix A for details on format). An LLM is used to read this text, and following a consistent prompt, it extracts four new categorical attributes for each task. This automated process converts the free text into a structured format that is suitable for further analysis. This newly structured dataset is then carried forward to the clustering stage.

In the second stage, the enriched data is used to group the maintenance tasks into clusters. The aim is to place tasks that share similar technical and failure characteristics into the same categories. This method differs from typical part classification, which often uses business data like cost or lead time. Instead, this clustering relies exclusively on the attributes of the tasks themselves, as presented in the MPD. Several different clustering algorithms are tested, and one is ultimately selected based on its ability to create sensible groupings for the model.

Once the tasks are clustered, each group is assigned a statistical model, which is initially a reasonable approximation but can be refined as more data becomes available. This is the aforementioned model used to estimate the probability that a part replacement will be needed when a given task from that cluster is performed. The statistical model itself is intentionally kept simple, mainly using a task's maintenance interval and its current age as inputs. This simplicity was chosen because there is no historical data available to properly validate a more complex model. The discussion also mentions other models that could be applied if such data were to become available in the future.

The derived failure probabilities are then integrated into an operations simulation. This simulation employs external software to map out a detailed flight and maintenance schedule for the fleet over a period of several years. The model's failure predictions are then applied to this operational schedule to estimate when part replacements might occur during regularly planned maintenance. The main output from this stage of the process is a time-stamped list identifying these potential replacements.

In the final step, all the individual part replacements predicted throughout the simulation are gathered together into a single dataset, forming the final, time-phased demand forecast for the planning period. To briefly recap, this forecast is informed by user inputs including a maintenance schedule, as well as NLP attributes, task clustering parameters, and task failure parameters. The process is depicted in Figure 3.1. The results section will present the format of this forecast and provide some basic descriptive statistics to summarize the output. This demand data is then used as the essential input for the procurement and inventory optimization analysis in the chapters that follow.

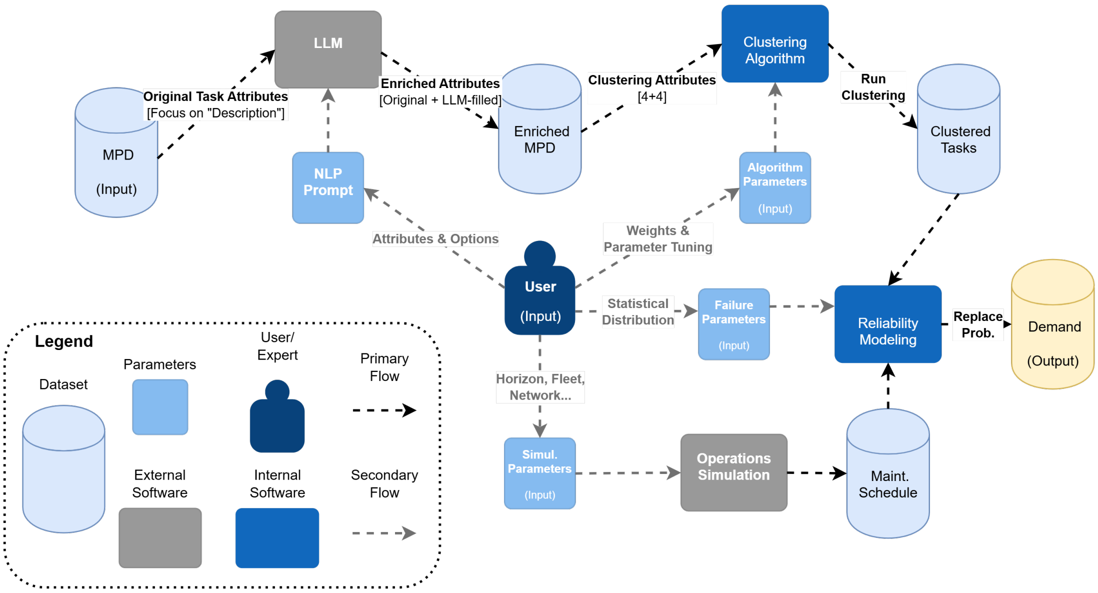


Figure 3.1: Process Flow for Generating a Synthetic Spare Part Demand Forecast.

3.1 Natural Language Processing Enrichment

The demand forecasting process begins by finding a way to get more useful information out of the MPD. While the MPD contains structured data like task intervals, ATA chapters, and man-hour estimates, it was not designed for spare parts management (see Appendix A). Key information needed to understand potential failures is often only present in the unstructured, free-text "Description" field of each task. When historical data or direct access to expert opinion is unavailable, the next best option is to systematically process the data that does exist. This section describes the method used to turn the free-text descriptions into structured attributes that can be used for modeling.

The challenge of using unstructured maintenance text is a recognized problem in the engineering field. Early research approached this using Natural Language Processing (NLP) techniques like topic modeling to pull features from maintenance logs for clustering and classification, though these methods had limits in understanding the full context of the text [Bokinsky et al., 2013; Usuga-Cadavid et al., 2022]. The field has since progressed towards models that better understand context, and recent studies show that LLMs can handle complex tasks like diagnosing faults from technical narratives [Nanyonga et al., 2025]. This thesis builds on this line of research, but uses the LLM in a slightly different way. Here, the LLM is not used to find a final answer like a fault diagnosis, but instead for feature engineering. Its role is to read the raw text in the MPD and generate new, structured categorical labels. This approach creates a solid, data-driven foundation for the main goal of using clustering to group tasks based on their shared technical characteristics.

In this study, feature engineering is performed using a commercial LLM, specifically the Meta Llama 3.3 70B model, which has been trained on a wide range of documents, including aviation-specific material. The process works by giving the LLM a prompt for each maintenance task. This prompt provides context, explains the goal, and presents a predefined set of four attributes, each with its own list of possible values. The LLM then reads the task's "Description" text and selects the single most appropriate value for each of the four attributes. This is done for every task in the MPD, creating a new, structured dataset that combines the original MPD data with the four new LLM-generated attributes.

Table 3.1: Attribute Description for Enriching MPD. Includes LLM's Possible Options.

Expected Failure Mode	Description: Dominant physical mechanism driving part replacement Options: Structural, Mechanical, Electrical, Leakage, Contamination or Blockage, No Failure, Not Applicable
Component Movement	Description: Whether primary part moves during normal operation Options: Moving, Intermittent Motion, Static, Not Applicable
Inspection Access Level	Description: Depth of access needed by technician Options: External Surface Only, Limited Internal Access, Full Internal Access, Disassembly Required, Not Applicable
Outcome Expectation	Description: Usual end state of task if item is serviceable Options: Visual Structural Continue, Functional Continue, Minor Servicing, Leakage or Contamination Continue, Replace Consumable, Replace Component, Overhaul or Bench Test, Repair On Aircraft, Not Applicable

These four generated attributes (see table 3.1) provide technical and operational context for each maintenance task. Two of the attributes focus on the component itself: Expected Failure Mode categorizes the

Table 3.2: Examples of Enriched MPD Tasks with LLM-Generated Attributes

Description	**Failure Mode**	**Movement**	**Access Level**	**Outcome Expectation**
APU bleed; Check valve remove and check auxiliary power unit bleed check valve for condition	Leakage	Moving	Disassembly Required	Functional Continue
Electrical power; Apply temporary protection to end fittings of electrical conduits in trimmable horizontal stabilizer	Electrical	Static	Limited Internal Access	Minor Servicing
Forward avionics compartment; General visual inspection of avionic compartment	No Failure	Static	External Surface Only	Visual Structural Continue

likely physical reason a part might need replacement, while Component Movement classifies whether the part is dynamic or static during normal operation. The other two attributes describe the maintenance activity. Inspection Access Level defines how intrusive the work is, from a surface-level check to requiring disassembly, and Outcome Expectation specifies the normal result of the task if no defects are found. The options for each attribute were carefully crafted to align with the content typically found in the "Description" column, ensuring relevance and applicability to the maintenance tasks at hand. However, it is acknowledged that the options may benefit from further refinement, potentially through collaboration with maintenance personnel who possess intimate knowledge of the tasks and procedures. Table 3.2 provides a sample, showing the attributes the LLM assigned to each task after processing its 'Description' text.

It's important to be clear about the limitations of this approach. Since there is no "ground truth" or expert-labeled data to compare against, the accuracy of the LLM's classifications cannot be mathematically proven. The choices for the attributes themselves are also based on logical reasoning rather than an established formula. However, for the purposes of this thesis, this method is a practical and sufficient way to enrich the dataset for the clustering that follows. Looking forward, this same technique could be a powerful tool if applied to other sources of free text, such as the maintenance notes and findings written by ground engineers, which would likely provide even more direct insight into component failures.

This LLM data-enriching process was applied to all maintenance tasks within this study's planning horizon (defined in Section 3.4). Table 3.3 presents the results, showing the percentage distribution of values assigned by the LLM for each of the four new attributes.

Table 3.3: Distribution of LLM-Generated Attribute Values

Attribute	**Value**	**Percentage of Tasks**
Expected Failure Mode	Structural	53.53%
	Mechanical	17.02%
	Electrical	12.87%
	Contamination or Blockage	8.11%
	Leakage	6.00%
	No Failure	1.94%
	Not Applicable	0.53%
Component Movement	Static	70.90%
	Moving	24.69%
	Not Applicable	3.09%
	Intermittent Motion	1.32%
Inspection Access Level	Limited Internal Access	46.38%
	External Surface Only	39.68%
	Not Applicable	7.23%
	Disassembly Required	3.70%
	Full Internal Access	3.00%
Outcome Expectation	Visual Structural Continue	59.79%
	Functional Continue	25.66%
	Minor Servicing	5.91%
	Leakage or Contamination Continue	4.06%
	Replace Consumable	3.35%
	Overhaul or Bench Test	0.44%
	Not Applicable	0.44%
	Replace Component	0.18%
	Repair On Aircraft	0.18%

The attribute distribution in the table reflects the source text's specific purpose. The MPD ”Description” column is written for ground engineers and primarily instructs them on how to approach an inspection. It generally does not detail complex repair procedures, as those are specified in separate manufacturer manuals. This focus on the task itself, rather than repairability, is why the generated attributes emphasize the inspection's nature (like 'Inspection Access Level' and 'Outcome Expectation').

Finally, while the accuracy of the LLM's output cannot be proven, its consistency is essential for building a repeatable framework. To test this, the enrichment process was run four additional times on the same MPD data. The consistency metric, shown in Table 3.4, measures how frequently all five runs assigned the identical value to an attribute for a given task. The results show a very high level of agreement, confirming that the LLM provides a stable and deterministic foundation for the subsequent analysis.

Table 3.4: LLM Output Consistency Across Five Runs Compared to First Run

Attribute	**5/5**	**4/5**	**3/5**	**2/5**	**1/5**
Expected Failure Mode	98.41%	0.71%	0.26%	0.35%	0.26%
Component Movement	96.03%	1.59%	0.62%	0.71%	1.06%
Inspection Access Level	89.07%	4.76%	1.85%	2.56%	1.76%
Outcome Expectation	98.77%	0.44%	0.53%	0.00%	0.26%

3.2 Clustering

With the data now enriched, the next step in the framework is to group maintenance tasks into clusters. The goal is to create groups where the tasks inside each cluster are expected to share a similar link between the maintenance action and the potential need for a spare part. This approach makes it more convenient to apply a consistent reliability model to all tasks that are grouped together.

The clustering is performed using a set of eight categorical attributes. Four of these are based on original MPD data: Source, Task Code, Zone, and Interval. These are used alongside the four attributes generated by the LLM in the previous step: Expected Failure Mode, Component Movement, Inspection Access Level, and Outcome Expectation. The selection of these eight attributes was based on the judgment that they best capture the technical and operational characteristics likely to influence whether a task leads to a component replacement.

To make the original data more suitable for this process, the many distinct values of the four source MPD attributes are first consolidated into a smaller number of broader categories. This mapping, detailed in Table 3.5, simplifies the dataset for the clustering algorithms by grouping the original data based on its practical meaning.

For instance, some Source codes in the MPD refer to mandatory tasks required for safety certification, so these are all put into a single ’certification’ group. In the same way, the many different Task Codes are simplified by grouping similar jobs together, such as putting all inspection-type tasks in one category and all servicing-type tasks in another. For the Zone and Interval data, tasks are grouped by their general scale; how many areas of the plane they affect or how many times they occur within this study’s planning horizon (see Section 3.4). This approach makes the data much simpler for the clustering algorithm to handle. For a description of the original codes, the reader is referred to the brief MPD overview in Appendix A.

Table 3.5: Mapping of Original MPD Attributes to Simplified Groups

Attribute	Original Values	Mapped Group
Source	`MRB 5, MRB 6, MRB 8`, etc.	Group 1: Safety or operational
	`CMR*, CMR**, ALI`	Group 2: ALS derived
	All other source values	Group 3: Other
Task Code	`DS, RS`	Group 1
	`CLN, LU, SV, CHK`	Group 2
	`FC, OP`	Group 3
	`DI, SDI, TPS`	Group 4
	`VC, GVI, BSI, RAR, OPT`, etc.	Group 5
Zone Count	1 zone listed	”1”
	2 zones listed	”2”
	3 or 4 zones listed	”3 or 4”
	6 or more zones listed	”6+”
Interval Frequency	Fewer than 1 occurrence per period	”<1 time”
	1 to 2 occurrences per period	”1 to 2 times”
	3 to 9 occurrences per period	”3 to 9 times”
	10 or more occurrences per period	”10+ times”

Algorithm Requirements

The prepared dataset is purely categorical, which places a key constraint on the selection of a clustering algorithm. Standard methods like k-means don’t work well in this situation, as they rely on calculating numerical distances between data points, a concept that isn’t meaningful for non-numeric data. This means the focus must shift to algorithms from the academic literature that are specifically designed to handle categorical attributes [Dinh et al., 2025]. These methods generally fall into a few different families, each with its own way of measuring similarity and forming groups.

One common choice is the K-modes algorithm, which is a version of the popular K-means method adapted for categorical data [Nguyen, 2017]. Instead of calculating averages for cluster centers, it uses the mode; the most frequent value for each attribute within a cluster. It groups tasks by minimizing the number of mismatched attributes. While fast, a known drawback is that its final result can depend on how the process starts, sometimes requiring multiple runs. A different, hierarchical approach is offered by the ROCK (RObust Clustering using linKs) algorithm [Guha et al., 2000]. Instead of just comparing two tasks directly, it first finds ”neighbors” for each task and then looks at how many common neighbors two tasks share; that count is the ”link”. The algorithm merges clusters in a way that maximizes these internal links, which helps find groups that are closely related. Another hierarchical method, Hierarchical Agglomerative Clustering (HAC), also builds groups from the bottom up [Müllner, 2011]. It starts by treating each data point as its own cluster and sequentially merges the most similar pair of clusters until all points form a single group. A fourth alternative, Latent Class Analysis (LCA), comes from statistics. This method assumes there are hidden (”latent”) groups in the data and uses a probabilistic model to determine which group each task most likely belongs to. A major advantage of LCA is that it provides statistical tests to help choose the best number of clusters [Sinha et al., 2021]. A comparative analysis of these methods is presented in Table 3.6.

Finally, it is useful to contrast the feature-based clustering methods discussed previously with traditional inventory classification schemes used in aviation MRO. These established techniques, such as ABC analysis, which prioritizes items by economic value, and VED analysis, which ranks parts by operational criticality (Vital, Essential, or Desirable), group parts based on their observed impact rather than their intrinsic technical attributes (see Section 2.1). While powerful for managing existing inventories, these methods are fundamentally retrospective. This highlights a key difference in approach: they are inventory management tools that analyze past performance, whereas this thesis requires a method that can generate a forecast proactively from a future maintenance plan.

Implementation

For this study, two hierarchical algorithms were selected for an initial implementation: Hierarchical Agglomerative Clustering (HAC) and ROCK. The selection was guided by the exploratory nature of the problem, where efficiency was not a concern due to the relatively modest dataset of 1,134 maintenance tasks. HAC offers the advantage of a dendrogram, which provides a visual guide for selecting a sensible number of clusters without a strong prior assumption. ROCK was chosen as a complementary method for its use of a more nuanced similarity measure based on shared neighbors, or ”links.” This approach has the potential to capture contextual relationships between tasks that direct attribute comparison might miss, which could be valuable for forming functionally related groups.

It is important to state that the implementation of these algorithms was not fully optimized. A true optimization would require iterative tuning of parameters and, crucially, the validation of the resulting clusters by subject matter experts, which is outside the scope of this study. The parameters were therefore chosen to provide a practical baseline for the analysis. However, it was observed that neither algorithm managed to produce clearly distinct or well-separated clusters (see next part), despite this being the intended purpose. With this limitation in mind, the clusters still provide a practical foundation for the reliability modeling step. The groupings are therefore treated as a preliminary, unvalidated classification.

The specific implementation and underlying logic of the HAC and ROCK algorithms are detailed below. For each method, a textual explanation of its process is provided, followed by a pseudocode that outlines the steps taken in this analysis. Given the preliminary nature of the clustering results, a selection of 7 (+1) clusters was chosen through trial and error, aiming for relatively uniform cluster sizes to facilitate further analysis. The selection of 7 clusters is deemed sufficient for this analysis, as further refinement of the clustering model is beyond the scope of this study. Moreover, this number strikes a balance between

Table 3.6: Comparative Analysis of Clustering Algorithms for Categorical Spare Parts Data

Feature	**K-Modes**	**ROCK (Robust Clustering using Links)**	**Latent Class Analysis (LCA)**	**Hierarchical Agglomerative Clustering (HAC)**
Core Principle	Partitional, dissimilarity-based. Aims to minimize the total dissimilarity between objects and their cluster's mode.	Hierarchical, link-based. Merges clusters based on the number of shared neighbors ("links"), capturing global structure.	Model-based, probabilistic. Assumes data is a mixture of distributions from several unobserved 'latent classes'.	Hierarchical, agglomerative. Builds a "bottom-up" hierarchy by successively merging the most similar pair of clusters.
Similarity Measure	Simple Matching Dissimilarity (Hamming Distance): The number of mismatched attributes between two objects.	Jaccard coefficient to define "neighbors", then counts shared neighbors to define "links" as the core merging criterion.	Model-based conditional probabilities. Assigns objects to the class where their response pattern is most likely.	A chosen distance metric (e.g., Gower's, Jaccard) for the initial dissimilarity matrix, combined with a linkage criterion (e.g., complete, average) to measure inter-cluster distance.
Cluster Representation	Mode: A vector of the most frequent categorical values for each attribute within the cluster.	A set of data points that have been agglomerated into a group. The cluster itself has no single 'center'.	A set of conditional response probabilities for each indicator, defining the 'profile' of a typical class member.	A dendrogram (tree structure) representing the full hierarchy of nested clusters. No single 'center' is defined.
Advantages	Computationally efficient and conceptually simple, facilitating rapid experimentation. Cluster representation via modes is highly intuitive and easy to interpret, clearly defining the profile of a typical maintenance task group.	Employs a sophisticated, context-aware similarity measure ("links") that can identify functionally related task groups missed by direct comparison. More robust to variations in cluster shape and density.	Provides a statistically rigorous, model-based framework for selecting the optimal number of clusters using fit indices (e.g., AIC, BIC). Produces probabilistic ("soft") cluster memberships, offering a more nuanced output.	Does not require the number of clusters to be specified a priori, allowing the data structure to guide the decision. The dendrogram output provides a powerful visual tool for exploring the data's hierarchical structure.
Disadvantages	Requires the number of clusters (K) to be specified beforehand, a choice that is often arbitrary and difficult to justify qualitatively. Highly sensitive to the random selection of initial modes, potentially leading to suboptimal solutions and lack of reproducibility across runs.	Iterative parameter tuning is time-consuming, even for moderately sized datasets. Performance is critically dependent on a non-intuitive similarity threshold parameter (θ), which is difficult to set without extensive experimentation.	Relies on the Local Independence Assumption, which is likely violated in maintenance data where attributes are correlated, potentially biasing results. The model-fitting algorithm can converge on local maxima, requiring multiple runs with different starting values to ensure a robust solution.	The "greedy" nature of the algorithm means early merge decisions are irreversible and cannot be corrected later. Highly sensitive to the choice of distance metric and linkage criterion.

diversity and coherence, avoiding the extremes of overly broad or excessively fragmented groupings, thereby providing a practical foundation for subsequent reliability modeling.

HAC

Hierarchical Agglomerative Clustering (HAC) is a ”bottom-up” algorithm that builds a hierarchy of clusters. The process begins by treating each individual maintenance task as its own cluster. It then computes a dissimilarity matrix containing the distance between every pair of tasks. For this study, the Hamming distance is used, which simply counts the number of attributes that differ between two tasks (x and y), as shown in the equation:

$$D_H(x, y) = \sum_{i=1}^{d} \mathbb{I}(x_i \neq y_i)$$

In an iterative process, the algorithm finds the two closest clusters in this matrix and merges them into a single new cluster. After each merge, the dissimilarity matrix is updated. The distance from the newly formed cluster to all other existing clusters is recalculated based on a chosen linkage criterion (in this case, ”weighted”). This merge-and-update cycle continues until only one large cluster containing all tasks remains. The full hierarchy, often visualized as a dendrogram, is then cut to produce the final number of clusters.

Algorithm 1. Hierarchical Agglomerative Clustering (HAC)

```
Input: A set of tasks $T = \{t_1, t_2, ..., t_n\}$; Number of desired clusters $N_{clusters}$
Output: A partition of tasks into $N_{clusters}$
Parameters chosen for this study:
$N_{clusters} \leftarrow 7$
Linkage Method ← 'weighted'
Distance Metric ← Hamming Distance
// Initialization
Let $C_i = \{t_i\}$ for $i = 1$ to $n$. The initial set of clusters is $\mathcal{C} = \{C_1, ..., C_n\}$
// Proximity Matrix Calculation
Compute the $n \times n$ dissimilarity matrix $D$, where $D(C_i, C_j)$ is the Hamming distance between the
  attribute vectors of the tasks in $C_i$ and $C_j$
// Iterative Merging
while number of clusters in $\mathcal{C} > 1$ do
    Find the two closest clusters, $C_i$ and $C_j$, in $\mathcal{C}$ based on the matrix $D$
    Merge the two clusters: $C_{new} \leftarrow C_i \cup C_j$
    Remove $C_i$ and $C_j$ from $\mathcal{C}$ and add $C_{new}$
    Update the dissimilarity matrix $D$ by calculating the distance from $C_{new}$ to all other remaining
      clusters using the chosen Linkage Method
// 4. Final Partitioning
Let $\mathcal{H}$ be the completed hierarchy of merges (the dendrogram)
Cut $\mathcal{H}$ at the level that results in exactly $N_{clusters}$
return The final partition of tasks into $N_{clusters}$
```

ROCK

The ROCK algorithm is a hierarchical method designed for categorical data and uses a link-based measure of similarity instead of direct distance. Two tasks are defined as neighbors if their similarity, measured here by the Jaccard coefficient, exceeds a threshold θ.

Let $S(t)$ be the set of active attribute–value indicators for task t (one per attribute, so $|S(t)| = 8$ in this study). Define the Jaccard similarity as $J(t_i, t_j) = \frac{|S(t_i) \cap S(t_j)|}{|S(t_i) \cup S(t_j)|}$. Tasks t_i and t_j are neighbors if $J(t_i, t_j) \geq \theta$ with $\theta = 0.05$; by definition $t_i \notin Neighbors(t_i)$.

The number of links between tasks t_i and t_j is given by:

$$link(t_i, t_j) = |Neighbors(t_i) \cap Neighbors(t_j)|.$$

This definition captures the structural context of each task by counting shared neighbors, rather than relying solely on pairwise similarity.

Goodness Measure. Clusters are merged according to a goodness measure that promotes high link density while penalizing merges that grow disproportionately large:

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{\epsilon} - n_i^{\epsilon} - n_j^{\epsilon}},$$

where $n_i = |C_i|$ and $n_j = |C_j|$. The exponent ϵ is defined as $\epsilon = 1 + 2f(\theta)$, where $f(\theta)$ determines how the expected number of links scales with cluster size at the chosen similarity threshold. So in a cluster of size n a typical point has about $n^{f(\theta)}$ neighbors, so the expected links scale as $n^{1+2f(\theta)}$.

Interpreting θ and ϵ. With Jaccard similarity $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, the neighbor rule implies:

$$|A \cap B| \geq \left\lceil \frac{\theta(|A| + |B|)}{1 + \theta} \right\rceil.$$

For $\theta = 0.05$ the neighbor condition requires at least $\left\lceil \frac{0.05\,(8+8)}{1+0.05} \right\rceil = 1$ shared attribute. The exponent $\epsilon = 2.46$ corresponds to $f(\theta) \approx 0.73$, consistent with the original ROCK formulation.

Algorithm 2. ROCK (RObust Clustering using linKs)

```
Input: Set of tasks $T = \{t_1, t_2, ..., t_n\}$; desired clusters $N_{clusters}$; threshold $\theta$; exponent $\epsilon$
Output: Partition of $T$ into $N_{clusters}$
Parameters used in this study:
$N_{clusters} \leftarrow 7$
$\theta \leftarrow 0.05$
$\epsilon \leftarrow 2.46$
Similarity function ← Jaccard coefficient
// Neighbor and Link Calculation
for each pair $(t_i, t_j)$ do
    if $J(t_i, t_j) \geq \theta$ then
        Add $t_j$ to $Neighbors(t_i)$ and $t_i$ to $Neighbors(t_j)$
for each pair $(t_i, t_j)$ do
    $link(t_i, t_j) \leftarrow |Neighbors(t_i) \cap Neighbors(t_j)|$
// Hierarchical Clustering using Goodness Measure
Initialize singleton clusters $C_i = \{t_i\}$; set $\mathcal{C} = \{C_1, ..., C_n\}$
Compute $g(C_i, C_j)$ for all pairs where $link[C_i, C_j] > 0$
while $|\mathcal{C}| > N_{clusters}$ do
    Select $(C_i, C_j)$ with highest $g(C_i, C_j)$
    Merge $C_{new} \leftarrow C_i \cup C_j$
    Update $\mathcal{C}$ and recompute $g$ values involving $C_{new}$
return $N_{clusters}$ final clusters
```

Clustering Results

The first step in analyzing the results is to isolate tasks that guarantee a spare part requirement. Maintenance tasks with the codes DS (Discard) and RS (Restore) are removed from the main dataset and assigned to a dedicated Cluster 1. The clustering algorithms are then applied to the remaining tasks with the objective of partitioning them into seven additional clusters. The resulting groupings are then analyzed both visually and quantitatively. For the HAC algorithm, a visual inspection of the dendrogram shows the cluster hierarchy, while for both algorithms, a Multidimensional Scaling (MDS) projection is used to visualize the clusters' structure.

For the HAC algorithm, the resulting hierarchy is visualized as a dendrogram (Figure 3.2). The diagram is truncated to focus on the main cluster branches, as a detailed analysis of the lower-level merges is beyond the scope of this thesis. The resulting clusters are color-coded for clarity, and the legend indicates the number of tasks within each group.

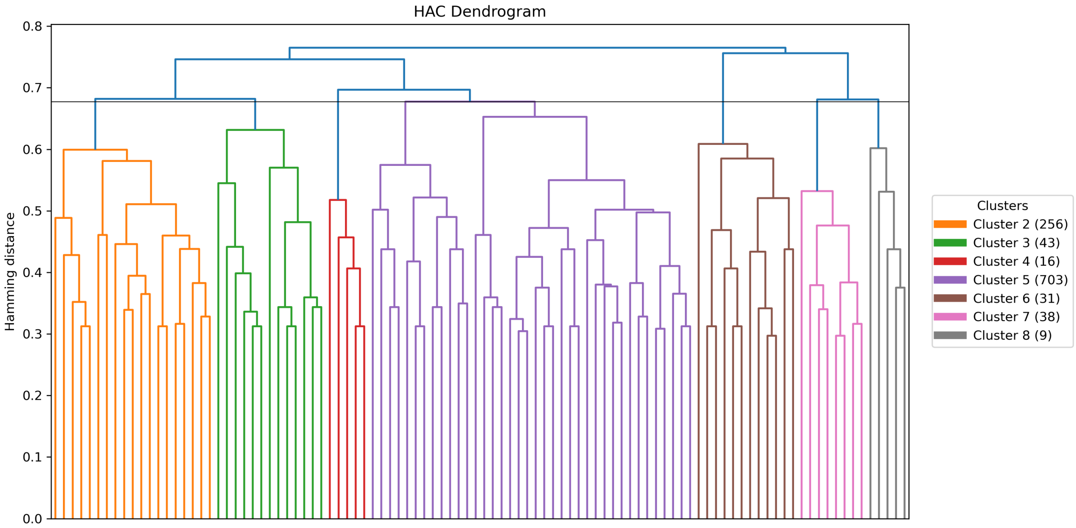


Figure 3.2: HAC Algorithm Dendrogram

Because it is impossible to visually plot the tasks in their original eight dimensions, the MDS projection is used to create a 2D scatter plot as a visual map of the data. The goal of MDS is simple: it arranges all the tasks on the plot in a way that honors their original dissimilarities. It uses the Hamming distance matrix as a guide, placing tasks that are very different far apart from each other, while positioning similar tasks close together. The clustering itself, however, is based on the original eight-dimensional data.

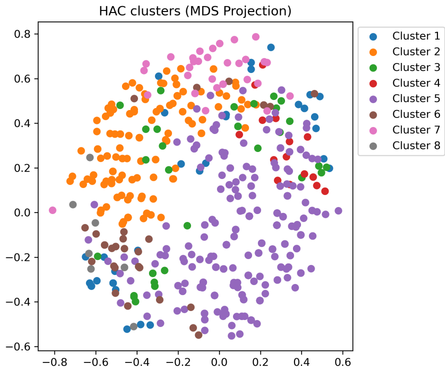


Figure 3.3: HAC Cluster Projection

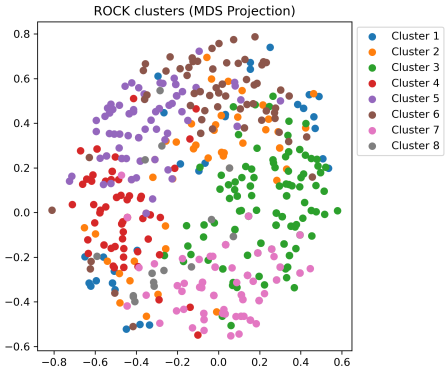


Figure 3.4: ROCK Cluster Projection

Table 3.7: Profiles of HAC Clusters (Top 3 Attribute Values)

Cluster (Size)	Source Group	Task Code Group	Zone Grp.	Interval Freq.	Failure Mode	Comp. Mvmt.	Access Lvl.	Outcome Expectation
1 (38)	Group 3: 50.0% Group 1: 50.0%	Group 1: 100.0%	1: 57.9% 2: 42.1%	1 to 2: 36.8% ¡1 time: 34.2% 3 to 9: 23.7%	Contam./Blockage: 76.3% Mechanical: 10.5% Electrical: 7.9%	Static: 84.2% Moving: 13.2% N/A: 2.6%	Disassembly Req.: 47.4% Limited Internal: 28.9% N/A: 15.8%	Replace Cons.: 71.1% Minor Servicing: 15.8% Overhaul/Bench Test: 7.9%
2 (256)	Group 3: 62.1% Group 1: 37.1% Group 2: 0.8%	Group 3: 86.3% Group 4: 7.0% Group 2: 4.7%	1: 79.7% 2: 11.3% 3 or 4: 7.4%	¡1 time: 52.0% 1 to 2: 34.4% 10+: 8.2%	Mechanical: 48.0% Electrical: 45.3% Leakage: 5.9%	Moving: 53.1% Static: 36.7% N/A: 7.0%	External Surface: 43.4% Limited Internal: 31.6% N/A: 23.0%	Funct. Cont.: 100.0%
3 (43)	Group 1: 72.1% Group 3: 27.9%	Group 5: 48.8% Group 4: 44.2% Group 3: 7.0%	1: 67.4% 2: 25.6% 3 or 4: 7.0%	¡1 time: 81.4% 1 to 2: 14.0% 3 to 9: 4.7%	Mechanical: 39.5% Structural: 23.3% Contam./Blockage: 16.3%	Moving: 69.8% Static: 27.9% N/A: 2.3%	Limited Internal: 39.5% Disassembly Req.: 37.2% Full Internal: 11.6%	Visual Struct. Cont.: 55.8% Replace Cons.: 23.3% Minor Servicing: 9.3%
4 (16)	Group 3: 100.0%	Group 4: 50.0% Group 2: 25.0% Group 5: 25.0%	3 or 4: 56.2% 2: 37.5% 1: 6.2%	3 to 9: 56.2% ¡1 time: 18.8% 1 to 2: 18.8%	Mechanical: 56.2% Structural: 37.5% Contam./Blockage: 6.2%	Moving: 75.0% Intermittent: 12.5% Static: 12.5%	Limited Internal: 93.8% Full Internal: 6.2%	Visual Struct. Cont.: 87.5% Minor Servicing: 6.2% Funct. Cont.: 6.2%
5 (703)	Group 3: 78.4% Group 2: 18.2% Group 1: 3.4%	Group 4: 54.5% Group 5: 39.3% Group 2: 4.3%	2: 55.3% 1: 33.3% 3 or 4: 7.4%	¡1 time: 70.7% 1 to 2: 18.8% 3 to 9: 5.4%	Structural: 83.9% Contam./Blockage: 5.7% Leakage: 4.7%	Static: 90.5% Moving: 8.1% N/A: 0.9%	Limited Internal: 52.9% External Surface: 42.8% Full Internal: 3.7%	Visual Struct. Cont.: 90.2% Leak/Contam. Cont.: 5.5% Minor Servicing: 3.0%
6 (31)	Group 1: 77.4% Group 3: 22.6%	Group 5: 51.6% Group 2: 25.8% Group 4: 19.4%	1: 67.7% 2: 32.3%	10+: 61.3% 3 to 9: 32.3% ¡1 time: 6.5%	Leakage: 45.2% Contam./Blockage: 25.8% Mechanical: 19.4%	Static: 80.6% Moving: 9.7% N/A: 6.5%	External Surface: 77.4% Limited Internal: 22.6%	Funct. Cont.: 61.3% Leak/Contam. Cont.: 19.4% Visual Struct. Cont.: 12.9%
7 (38)	Group 1: 68.4% Group 3: 31.6%	Group 2: 100.0%	2: 52.6% 1: 34.2% 3 or 4: 13.2%	1 to 2: 39.5% ¡1 time: 26.3% 3 to 9: 23.7%	Mechanical: 76.3% Contam./Blockage: 13.2% N/A: 10.5%	Moving: 94.7% N/A: 5.3%	Limited Internal: 60.5% N/A: 31.6% External Surface: 7.9%	Minor Servicing: 81.6% Funct. Cont.: 10.5% Visual Struct. Cont.: 5.3%
8 (9)	Group 3: 100.0%	Group 2: 88.9% Group 3: 11.1%	1: 88.9% 2: 11.1%	10+: 66.7% 1 to 2: 22.2% 3 to 9: 11.1%	Electrical: 55.6% N/A: 22.2% Contam./Blockage: 11.1%	N/A: 55.6% Static: 33.3% Moving: 11.1%	External Surface: 77.8% N/A: 22.2%	N/A: 55.6% Minor Servicing: 22.2% Funct. Cont.: 11.1%

Note on Abbreviations: N/A = Not Applicable; Contam. = Contamination; Funct. Cont. = Functional Continue; Struct. = Structural; Comp. Mvmt. = Component Movement; Access Lvl. = Access Level; Disassembly Req. = Disassembly Required; Leak/Contam. Cont. = Leakage or Contamination Continue; Visual Struct. Cont. = Visual Structural Continue; Replace Cons. = Replace Consumable.

Table 3.8: Profiles of ROCK Clusters (Top 3 Attribute Values)

Cluster (Size)	Source Group	Task Code Group	Zone Grp.	Interval Freq.	Failure Mode	Comp. Mvmt.	Access Lvl.	Outcome Expectation
1 (38)	Group 3: 50.0% Group 1: 50.0%	Group 1: 100.0%	1: 57.9% 2: 42.1%	1 to 2: 36.8% ¡1 time: 34.2% 3 to 9: 23.7%	Contam./Blockage: 76.3% Mechanical: 10.5% Electrical: 7.9%	Static: 84.2% Moving: 13.2% N/A: 2.6%	Disassembly Req.: 47.4% Limited Internal: 28.9% N/A: 15.8%	Replace Cons.: 71.1% Minor Servicing: 15.8% Overhaul/Bench Test: 7.9%
2 (58)	Group 3: 65.5% Group 1: 34.5%	Group 2: 55.2% Group 3: 24.1% Group 4: 19.0%	1: 53.4% 2: 44.8% 3 or 4: 1.7%	¡1 time: 48.3% 1 to 2: 25.9% 10+: 13.8%	Contam./Blockage: 62.1% Leakage: 22.4% Electrical: 12.1%	Static: 87.9% Moving: 8.6% N/A: 3.4%	Limited Internal: 75.9% External Surface: 15.5% Disassembly Req.: 5.2%	Leak/Contam. Cont.: 50.0% Funct. Cont.: 22.4% Minor Servicing: 22.4%
3 (473)	Group 3: 83.3% Group 2: 10.6% Group 1: 6.1%	Group 4: 79.5% Group 5: 20.1% Group 3: 0.2%	2: 58.8% 1: 26.2% 3 or 4: 10.4%	¡1 time: 82.2% 1 to 2: 12.1% 3 to 9: 4.4%	Structural: 87.1% Mechanical: 3.4% Electrical: 3.2%	Static: 85.4% Moving: 13.1% Intermittent: 0.8%	Limited Internal: 71.7% External Surface: 22.2% Full Internal: 5.9%	Visual Struct. Cont.: 96.0% Minor Servicing: 2.1% Funct. Cont.: 1.1%
4 (145)	Group 3: 79.3% Group 1: 20.7%	Group 3: 83.4% Group 5: 9.0% Group 2: 6.2%	1: 94.5% 2: 4.8% 3 or 4: 0.7%	¡1 time: 35.2% 1 to 2: 31.7% 10+: 24.8%	Electrical: 73.1% Leakage: 11.7% Mechanical: 10.3%	Static: 72.4% N/A: 15.2% Moving: 11.0%	External Surface: 75.2% N/A: 20.0% Limited Internal: 3.4%	Funct. Cont.: 89.7% Visual Struct. Cont.: 3.4% Leak/Contam. Cont.: 3.4%
5 (126)	Group 1: 62.7% Group 3: 36.5% Group 2: 0.8%	Group 3: 77.8% Group 4: 11.9% Group 2: 9.5%	1: 69.8% 3 or 4: 15.9% 2: 11.1%	¡1 time: 65.1% 1 to 2: 24.6% 10+: 5.6%	Mechanical: 82.5% Electrical: 10.3% Leakage: 7.1%	Moving: 88.9% Intermittent: 5.6% Static: 4.8%	Limited Internal: 48.4% N/A: 25.4% External Surface: 23.0%	Funct. Cont.: 93.7% Visual Struct. Cont.: 4.0% Overhaul/Bench Test: 1.6%
6 (68)	Group 3: 52.9% Group 1: 47.1%	Group 2: 67.6% Group 4: 13.2% Group 5: 11.8%	2: 64.7% 1: 23.5% 3 or 4: 11.8%	1 to 2: 39.7% 3 to 9: 27.9% ¡1 time: 23.5%	Mechanical: 73.5% Contam./Blockage: 10.3% N/A: 8.8%	Moving: 89.7% N/A: 5.9% Intermittent: 2.9%	Limited Internal: 66.2% N/A: 17.6% External Surface: 10.3%	Minor Servicing: 50.0% Funct. Cont.: 30.9% Visual Struct. Cont.: 13.2%
7 (206)	Group 3: 59.7% Group 2: 38.3% Group 1: 1.9%	Group 5: 89.8% Group 4: 9.7% Group 3: 0.5%	2: 47.1% 1: 45.6% 3 or 4: 4.4%	¡1 time: 46.1% 1 to 2: 33.5% 10+: 11.7%	Structural: 92.7% Leakage: 2.9% Mechanical: 1.5%	Static: 92.2% Moving: 6.8% N/A: 1.0%	External Surface: 91.3% Limited Internal: 8.7%	Visual Struct. Cont.: 96.6% Leak/Contam. Cont.: 1.9% Funct. Cont.: 1.5%
8 (20)	Group 3: 70.0% Group 1: 30.0%	Group 5: 95.0% Group 4: 5.0%	1: 100.0%	¡1 time: 95.0% 1 to 2: 5.0%	Contam./Blockage: 40.0% Leakage: 30.0% No Failure: 25.0%	Static: 75.0% Moving: 25.0%	Disassembly Req.: 65.0% Limited Internal: 15.0% N/A: 15.0%	Replace Cons.: 50.0% Visual Struct. Cont.: 20.0% Minor Servicing: 15.0%

Note on Abbreviations: N/A = Not Applicable; Contam. = Contamination; Funct. Cont. = Functional Continue; Struct. = Structural; Comp. Mvmt. = Component Movement; Access Lvl. = Access Level; Disassembly Req. = Disassembly Required; Leak/Contam. Cont. = Leakage or Contamination Continue; Visual Struct. Cont. = Visual Structural Continue; Replace Cons. = Replace Consumable.

An inspection of the results (Tables 3.7 & 3.8) from both algorithms confirms that the clustering is not random and produces thematically coherent groups. The clusters generally form around a dominant value in one or two key attributes, giving each a distinct profile. For example, a clear ”Structural Inspection” cluster emerges in both results (HAC Cluster 5, ROCK Cluster 3), defined by a high concentration of the Structural failure mode and the Visual Structural Continue outcome. Similarly, both methods identify a distinct ”Mechanical Systems” cluster (HAC Cluster 7, ROCK Cluster 5). A notable difference is that the ROCK algorithm produced more uniformly sized large clusters, whereas the HAC results are more skewed, containing one very large cluster (703 tasks) and several very small ones (e.g., Cluster 8 with 9 tasks).

However, with eight different categorical attributes of equal weight, it is challenging for these unsupervised algorithms to create perfectly distinct clusters that clearly map to specific component failure behaviors. The high dimensionality means that while broad themes are captured, there is still significant variation within each cluster. A more refined approach involving attribute weighting or expert validation would be needed to create a definitive taxonomy. Nevertheless, for the exploratory purposes of this thesis, these groupings provide a sufficient and practical foundation to proceed to the next stage of the framework: assigning reliability models to each cluster to estimate failure probabilities.

3.3 Reliability Modeling

After grouping the maintenance tasks, the next step is to assign a reliability model to each cluster. The purpose of this model is to estimate the probability that performing a task from a given cluster will result in a spare part replacement. This approach provides a practical way to handle the failure characteristics of many different tasks efficiently. The quality of this step naturally depends on the coherence of the task groupings created by the clustering algorithms. It is important to assign a unique model to each cluster. Applying a single, universal failure probability to all tasks would be an inaccurate and overly simplistic assumption, failing to capture the distinct reliability profiles that different component and task groups are expected to have.

This section proceeds in two parts. The first, an overview of applicable reliability models, discusses several statistical methods from the literature that are suitable for modeling component failure. Following this, the specific reliability model adopted for the analysis is presented, including a detailed description of its mechanics and its complete mathematical formulation.

3.3.1 Applicable Models

To estimate the probability that a maintenance task will lead to a component replacement, several families of statistical models can be used. The most suitable choice often depends on the type of data available and the specific failure characteristics of the components being studied.

A foundational approach in reliability engineering is the use of parametric survival models, with the Weibull distribution being a common choice. This model is defined by a shape parameter (β) and a scale parameter (α). The value of β is particularly useful as it offers insight into the component's life cycle: $\beta < 1$ suggests early-life failures (infant mortality), $\beta = 1$ indicates a constant failure rate, and $\beta > 1$ points to wear-out failures that increase with age. By fitting this distribution to historical failure data, it is possible to model the time-to-failure for a group of components [Kontrec et al., 2016].

The reliability of a component, however, is often influenced by external factors beyond its age. More advanced models can incorporate this extra information, known as covariates, to improve predictive accuracy. The Cox Proportional Hazards Model (PHM), for example, is a regression model that can assess the effect of factors like flight hours, operational environment, or component manufacturer on the failure rate. It is particularly effective at handling censored data, which is common in maintenance records where many components are inspected but do not fail [Verhagen and De Boer, 2018]. A more direct approach is to use the maintenance plan itself as a form of Advance Demand Information (ADI). One such method uses a Binomial distribution to model the number of replacements arising from a known number of scheduled tasks. For a given period, the demand is estimated as $B(A_t, p)$, where A_t is the number of planned tasks and p is the historical probability that a task leads to a replacement. This technique is practical because it does not require complex component degradation data [Zhu et al., 2020].

An alternative to modeling time-to-failure is to frame the problem as a binary classification task: for each maintenance check, predict whether a component will be replaced (1) or not (0). Logistic Regression is a fundamental model for this purpose, offering clear, interpretable results. For more complex, non-linear relationships, ensemble methods like Random Forests or Gradient Boosting Machines often provide higher accuracy. A known drawback of these models is their lack of transparency, often being referred to as ”black boxes.” However, this can be addressed with techniques from Explainable AI (XAI), such as SHAP or LIME, which help to interpret the model's predictions [te Booij, 2024].

3.3.2 Proposed Reliability Model

The two-parameter Weibull distribution is selected to model the probability of a component replacement. This model is defined by a scale parameter, α, and a shape parameter, β. To ensure a clear and consistent mathematical foundation, this analysis adopts the standard formulation for these parameters. The key functions and their notation, as detailed below, follow the representation used in the Springer Handbook of Engineering Statistics [Pham, 2006].

The **Cumulative Distribution Function (CDF)**, $F(t)$, gives the probability that a component fails at or before age t. This is the primary function used to calculate the probability of a replacement during a check.

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right], \quad t \geq 0$$

The **Reliability Function**, $R(t)$, is the complement of the CDF and gives the probability that a component will survive beyond age t.

$$R(t) = \exp\left[-\left(\frac{t}{\alpha}\right)^{\beta}\right]$$

The **Failure Rate Function**, $h(t)$, also known as the hazard rate, gives the instantaneous risk of failure at age t, assuming the component has survived until that point.

$$h(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}$$

The shape parameter β is particularly important as it describes the component's failure characteristic over its life. A value of $\beta < 1$ indicates a decreasing failure rate (infant mortality), $\beta = 1$ indicates a constant rate (random failures), and $\beta > 1$ indicates an increasing rate (wear-out failures).

To apply this model, the parameters must be specified for each maintenance task. The approach for this study is as follows:

- **Scale Parameter (α):** This parameter represents the characteristic life of the component. It is the time at which approximately 63.2% of the population is expected to have failed. In this study, α is set as the scheduled maintenance interval for the task, which is measured in Flight Hours (FH), Flight Cycles (FC), or Days (DY). This directly links the model to the operational plan.
- **Shape Parameter (β):** This parameter is user-driven. As no historical data is available for estimation, shape parameters are assigned incrementally to simulate different wear-out rates: $\beta = 1.1$ for Cluster 2, increasing by 0.1 for each subsequent group up to $\beta = 1.7$ for Cluster 8. This range reflects an assumption of a wear-out failure process, which is common for many mechanical and electrical components. Components that degrade over time may have higher β values, while those with relatively constant failure rates may have β values around 1. Components prone to early-life failures may have lower β values.

- **Time Variable (t):** This input represents the component's age at the time of the check. Its unit matches that of the scale parameter α: Flight Hours (FH), Flight Cycles (FC), or Days (DY). To account for potential unscheduled failures occurring between planned checks, the age input can be adjusted by a constant factor. For this study, the age input was scaled by dividing it by three. This adjustment factor is a preliminary assumption chosen to generate realistic failure probabilities, and it would need to be properly tuned using real-world data.
- **Mandatory Replacements:** Tasks grouped in **Cluster 1** are by definition mandatory replacements (e.g., discard or restore tasks). They are assigned a failure probability of 1.0 and are not evaluated with the Weibull model.

The two-parameter Weibull model has a failure rate that is monotonic, meaning it can only increase, decrease, or remain constant. This is a limitation if failure data exhibits more complex behavior, such as a ”bathtub” curve which includes high infant mortality, a low random failure rate, and a high wear-out rate. If historical data indicated such a pattern, the framework could be extended to use more advanced models. For example, the handbook describes several generalized Weibull distributions, such as the Modified Weibull Distribution, which can model bathtub-shaped failure rates for certain parameter values [Pham, 2006, p. 71]. Furthermore, with a historical dataset, the suitability of the Weibull model itself could be statistically verified using goodness-of-fit tests like the Anderson-Darling or Kolmogorov-Smirnov test [Pham, 2006, p. 69].

3.4 Operations Simulation

Flying Schedule

The operations simulation is based on a fleet of eight aircraft: six Airbus A318-112, one Airbus A319-115, and one Airbus A320-233. These aircraft operate on a network of routes connecting several European airports. In the Netherlands, these include Amsterdam (AMS), Eindhoven (EIN), Maastricht (MST), and Rotterdam (RTM). The network also serves two airports in London, Heathrow (LHR) and Stansted (STN), and two in Paris, Charles de Gaulle (CDG) and Orly (ORY), along with Antwerp (ANR) in Belgium and Hamburg (HAM) in Germany.

This flight schedule is generated to reflect realistic operational patterns over a planning horizon of five years, from January 1, 2026, to December 31, 2030. The primary concern for this analysis is the resulting schedule of aircraft utilization, which provides the necessary input for the maintenance plan. As the focus is on the output rather than the generation process itself, the internal mechanics of the scheduling algorithm are considered outside the scope of this discussion. Table 3.9 shows a sample of the resulting flight log for a single aircraft, detailing its status and cumulative usage over a short period.

Maintenance Schedule

Following the flight operations, the simulation generates a maintenance schedule by grouping individual tasks into packages. This approach simplifies the scheduling problem by first creating pre-defined work packages, which are then assigned to specific maintenance opportunities. The main logic for grouping tasks is based on their maintenance intervals and the physical zones on the aircraft that technicians must access.

A core component of this strategy is the creation of equalized work packages. This method differs significantly from conventional maintenance programs, which group tasks into large, infrequent ”letter checks” (e.g., A-checks, C-checks) that take an aircraft out of service for several days. Instead, the equalized package approach breaks the entire maintenance program into much smaller, more frequent blocks. Each package is ”equalized” to contain a similar, manageable amount of work, measured in man-hours.

Table 3.9: (Sample) Flight Log for a Single Aircraft A318-112

Date	**Time**	**Event**	**Location**	**Cum. FC**	**Cum. FH**
13-01-2026	18:25	Taxi and Take-Off	EIN	13934	29733.42
13-01-2026	18:30	In Flight	EIN		
13-01-2026	19:40	Landing and Taxi	HAM	13935	29734.42
13-01-2026	19:45	Parked at Gate	HAM		
13-01-2026	20:05	Taxi and Take-Off	HAM		
13-01-2026	20:10	In Flight	HAM		
13-01-2026	21:20	Landing and Taxi	EIN	13936	29735.42
13-01-2026	21:25	Parked at Gate	EIN		
13-01-2026	21:30	Taxi to Maintenance or Parking	EIN		
13-01-2026	21:40	Maintenance or Parked	EIN		
14-01-2026	07:00	Taxi to Gate	EIN		
14-01-2026	07:10	Parked at Gate	EIN		
14-01-2026	07:15	Taxi and Take-Off	EIN		
14-01-2026	07:20	In Flight	EIN		
14-01-2026	07:30	Landing and Taxi	STN	13937	29735.58
14-01-2026	07:35	Parked at Gate	STN		
14-01-2026	07:55	Taxi and Take-Off	STN		
14-01-2026	08:00	In Flight	STN		
14-01-2026	10:05	Landing and Taxi	EIN	13938	29737.33

The purpose of these smaller packages is to fit necessary maintenance into the short, overnight ground slots that occur naturally in an aircraft's flight schedule. Unlike traditional heavy checks that provide a multi-day buffer to source missing items, these tight overnight windows preclude reactive procurement, requiring parts to be immediately available. For the purposes of this study, each package is designed to be completed within an available capacity of 26 man-hours, with any night being a potential candidate for base maintenance. By performing maintenance in these small, regular increments, the need for long, dedicated downtimes is minimized, which in turn increases overall aircraft availability and operational flexibility. The scheduling software also has the ability to slightly adjust a task's interval within its allowed limits to build more efficient packages, with the ultimate goal of minimizing the total number of maintenance visits for the fleet. For this study, an interval exceedance of up to 3.5% is permitted.

Tables 3.10 and 3.11 provide a specific example of this process. The schedule in Table 3.10 lists the packages performed during the overnight ground time of the aircraft whose flight activity was shown in Table 3.9. Table 3.11 provides the corresponding details for these packages, showing the specific MPD tasks they contain and their respective intervals.

Table 3.10: Maintenance Schedule (Sample)

Date	**Start Time**	**End Time**	**Package Code**	**Location**
13-01-2026	00:21	01:21	143	EIN
13-01-2026	00:21	01:21	89	EIN
13-01-2026	00:21	01:33	28	EIN
13-01-2026	01:15	01:39	136	EIN
13-01-2026	01:21	01:33	22	EIN
13-01-2026	01:33	01:45	25	EIN
13-01-2026	02:00	02:01	137	EIN
13-01-2026	02:01	02:03	134	EIN
13-01-2026	02:03	02:03	133	EIN
13-01-2026	02:03	03:03	142	EIN
13-01-2026	02:33	03:57	144	EIN
13-01-2026	03:03	03:27	7	EIN
13-01-2026	03:15	03:39	149	EIN
13-01-2026	04:00	04:12	139	EIN

Table 3.11: Maintenance Package Definitions (A318-112)

Package Code	Interval	Tasks Included
134	150 FH OR NOTE 5	242100-02-1
22	720 FC	523301-01-1
133	120 FH	242400-01-2
139	1 A	272651-01-1
142	2 A OR 500 FC NOTE 9	321000-01-1, 322000-01-1
143	2 A	212643-09-1, 321111-03-1, 783100-C1-1, 316322-02-1, ZL-129-01-2, 801110-C3-1, 290000-05-1, 535900-01-1, 383200-01-1, 290000-02-1, 212300-02-1, 256241-05-1, 212643-02-1, 324311-01-1, 722000-C1-1, ZL-129-01-1
144	2 A OR NOTE 7	311111-C1-1
25	1500 FC	534108-01-6, 531111-01-1
136	500 FH NOTE 21	282800-09-1, 261700-03-1, 242800-01-2, 261600-04-1, 256600-02-1, 271400-01-2, 261600-03-1, 255000-12-1, 255000-11-1, 273400-01-2, 255000-01-1
137	500 FH OR NOTE 3	211119-02-1
149	4 A	290000-04-1
28	1600 FC OR NOTE 7	250000-C1-3, 724000-C1-1
37	4800 FC	572013-01-2

3.5 Results: Spare Part Demand

Applying the reliability models to the operations simulation produces the time-stamped spare part demand forecast that is the final output of this chapter. A key assumption is made to handle the ”cold start” of the simulation: the first time a task is performed within the planning horizon on any aircraft, its component’s age is assumed to be 100% of its scheduled interval. The influence of this initial condition diminishes over the five-year planning horizon as the maintenance schedule develops its own history. After the first check, component age is tracked based on the time since its last completion.

An example for a single A318-112 aircraft is shown in Figure 3.5 to illustrate the model’s output. It lists the scheduled dates and calculated failure probabilities for three tasks: 242400-01-2 (120 FH), 321119-02-1 (500 FH), and 242100-03-1 (800 FH). The variation in probabilities for the same task over time shows the dynamic nature of the Weibull model, where the probability depends directly on the component’s age. For the mandatory replacement (task code: DS,RS), this probability is fixed at 1.

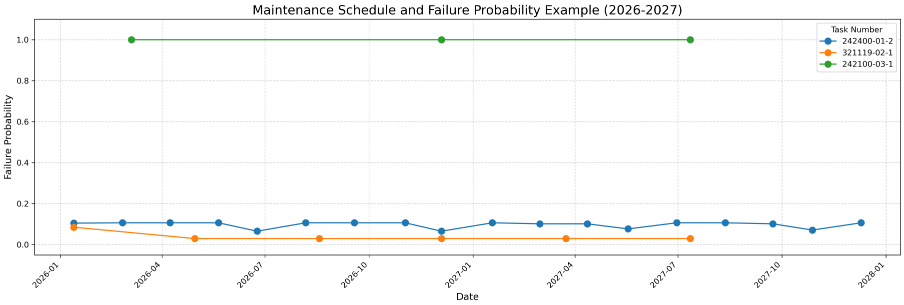


Figure 3.5: Maintenance Schedule and Failure Probability for Three Tasks; A318-112 (2026-2027).

Table 3.12 presents the aggregated monthly demand forecast for the entire fleet across all tasks, broken down by task code. Each cell shows the total expected number of replacements for that category, calculated by summing the failure probabilities of all tasks scheduled for that month. The fluctuations in expected demand are a direct reflection of the varying number of maintenance tasks scheduled in the underlying operational plan. These figures are stochastic estimates, not deterministic counts, and serve as a probabilistic guide for planning.

Table 3.12: Monthly Aggregate Spare Part Demand by Task Code (2026-2030)

TASK CODE	BSI	CHK	CLN	DI	DS	FC	GVI	LU	OP	OPT	RAR	RS	SDI	SV	TPS	VC	Total
Month																	
2026-01	0.70	12.85	1.65	21.79	11.00	1.72	38.36	3.17	49.02	0.19	0.47	26.00	0.58	2.76	0.00	19.24	189.50
2026-02	0.23	4.80	0.19	7.03	11.00	3.00	8.05	2.06	6.96	0.00	1.88	2.00	0.62	2.11	0.00	0.88	50.82
2026-03	0.00	4.30	0.39	7.44	8.00	3.98	6.24	1.57	17.35	0.00	0.47	2.00	0.00	0.46	0.00	4.22	56.41
2026-04	0.00	3.16	0.00	5.55	0.00	0.52	9.08	0.65	15.92	0.00	0.00	8.00	0.00	0.36	0.00	4.32	47.57
2026-05	0.58	8.37	0.58	11.09	2.00	1.41	23.12	1.65	34.83	0.00	0.01	12.00	0.00	1.12	0.58	10.50	107.84
2026-06	1.14	2.27	0.51	4.17	9.00	1.29	8.10	2.26	18.76	0.00	0.01	7.00	0.52	0.53	0.97	3.32	59.84
2026-07	0.19	5.15	0.15	10.21	6.00	1.64	17.14	0.73	22.71	0.00	0.01	13.00	0.20	0.61	0.97	5.90	84.61
2026-08	0.62	5.16	0.44	6.42	4.00	0.87	10.70	0.88	21.50	0.00	0.01	7.00	0.97	0.37	0.00	5.09	64.01
2026-09	0.00	5.71	0.60	7.34	7.00	2.90	16.06	1.86	23.75	0.19	0.01	10.00	0.19	0.83	0.77	6.62	83.84
2026-10	0.00	7.48	0.00	16.11	10.00	2.64	28.42	3.36	42.98	0.43	0.77	17.00	1.16	1.74	1.74	7.54	141.37
2026-11	0.00	2.88	0.33	7.52	5.00	1.58	9.68	1.04	17.13	0.00	0.00	4.00	0.97	0.51	0.77	2.69	54.09
2026-12	0.21	8.56	2.11	22.74	18.00	2.94	47.06	8.91	55.79	0.00	1.28	16.00	0.36	5.82	5.99	9.65	205.43
2027-01	0.48	6.18	0.35	10.07	5.00	1.64	17.33	0.97	20.95	0.00	0.00	12.00	0.74	1.20	0.00	6.86	83.76
2027-02	0.32	5.67	0.32	7.51	5.00	0.90	15.59	1.38	21.71	0.00	0.02	6.00	1.32	0.80	0.00	4.90	71.44
2027-03	0.33	2.72	0.19	4.20	4.00	1.18	6.22	0.86	12.47	0.00	0.00	6.00	0.33	0.54	0.00	3.22	42.27
2027-04	0.11	2.89	0.21	7.51	2.00	1.30	6.12	1.30	11.72	0.00	0.01	2.00	0.22	0.48	0.00	4.04	39.91
2027-05	0.12	9.35	0.84	13.84	8.00	1.81	29.51	1.93	32.59	0.19	0.01	16.00	0.52	1.75	0.00	10.96	127.42
2027-06	0.12	3.57	0.38	4.76	5.00	1.93	7.62	1.10	10.88	0.00	0.25	4.00	0.68	0.62	0.00	3.14	44.05
2027-07	0.00	4.93	0.00	8.55	5.00	2.51	14.21	0.00	23.88	0.00	0.24	9.00	0.60	0.35	0.00	6.21	75.47
2027-08	0.00	2.00	0.38	4.45	2.00	1.44	7.43	0.94	11.98	0.00	0.01	6.00	0.39	0.24	0.24	2.42	39.92
2027-09	0.61	7.69	0.19	12.74	4.00	2.42	22.85	1.49	32.33	0.00	0.01	12.00	1.10	1.13	0.23	10.08	108.87
2027-10	0.00	3.94	0.19	7.86	6.00	2.55	9.89	0.72	15.88	0.00	0.02	10.00	0.58	0.77	0.13	3.98	62.50
2027-11	0.58	4.08	0.26	9.70	4.00	1.75	12.74	1.36	19.56	0.00	0.24	8.00	0.37	0.32	0.00	5.42	68.38
2027-12	0.00	2.28	0.08	5.79	5.00	0.54	8.02	0.59	15.66	0.00	0.01	6.00	0.77	0.35	0.39	2.90	48.37
2028-01	0.72	6.46	0.74	9.45	8.00	1.52	14.78	1.57	21.55	0.19	0.01	13.00	0.59	1.05	0.00	8.21	87.85
2028-02	0.14	3.85	0.00	8.00	4.00	1.44	14.51	0.49	14.53	0.00	0.01	5.00	1.31	0.50	0.00	4.26	58.03
2028-03	0.21	4.82	0.63	15.55	8.00	2.67	17.85	2.10	28.83	0.00	0.00	10.00	1.73	0.89	0.44	6.54	100.27
2028-04	0.24	5.58	0.35	11.09	3.00	2.27	16.56	0.97	18.61	0.00	0.25	11.00	0.60	0.73	0.00	4.66	75.92
2028-05	0.10	5.90	0.71	10.99	5.00	2.80	13.62	2.07	25.17	0.00	0.48	9.00	0.16	1.85	0.00	6.40	84.25
2028-06	0.00	6.29	0.73	15.14	5.00	1.12	19.85	1.14	24.87	0.00	0.02	12.00	0.21	1.10	0.00	8.23	95.71
2028-07	0.37	3.18	0.09	14.04	10.00	4.87	13.61	2.27	22.97	0.00	0.15	9.00	0.75	1.45	0.79	3.02	86.56
2028-08	0.34	6.82	0.95	32.80	38.00	10.14	29.99	5.05	67.64	0.24	2.66	18.00	3.89	2.73	3.02	5.65	227.91
2028-09	0.39	10.05	0.54	20.76	5.00	1.07	27.41	1.79	36.98	0.19	0.02	19.00	0.83	1.78	0.00	12.87	138.68
2028-10	0.20	4.59	0.19	8.85	3.00	0.92	7.23	1.02	12.84	0.00	0.00	4.00	0.92	0.66	0.00	2.79	47.20
2028-11	0.23	5.59	0.54	13.16	9.00	2.36	14.48	1.11	23.31	0.00	0.02	12.00	1.08	0.96	0.23	8.13	92.17
2028-12	0.16	2.88	0.39	8.16	2.00	0.69	9.33	0.83	14.01	0.00	0.01	7.00	0.96	0.38	0.22	2.83	49.83
2029-01	0.33	3.94	0.19	14.85	3.00	1.75	13.06	1.51	18.41	0.00	0.00	5.00	0.42	0.65	0.10	5.08	68.29
2029-02	0.19	4.74	0.53	17.72	2.00	0.89	15.47	0.54	22.92	0.00	0.01	11.00	0.97	0.48	0.00	7.60	85.05
2029-03	0.14	2.98	0.25	10.77	3.00	0.86	8.43	0.51	8.66	0.00	0.01	2.00	0.49	0.29	0.00	1.52	39.91
2029-04	0.00	3.18	0.19	8.01	1.00	0.66	13.95	1.36	17.90	0.00	0.01	6.00	0.00	0.30	0.19	5.03	57.79
2029-05	0.22	7.14	0.59	12.52	12.00	2.57	20.93	1.48	29.38	0.19	0.01	15.00	0.77	1.55	0.20	9.37	113.95
2029-06	0.00	5.12	1.09	16.66	4.00	1.43	15.24	1.03	18.67	0.00	0.00	14.00	1.96	0.80	0.00	6.47	86.48
2029-07	0.56	4.16	0.09	18.39	7.00	2.54	12.69	1.47	16.45	0.00	0.02	4.00	1.18	1.14	0.37	3.50	73.56
2029-08	0.48	2.45	0.00	16.04	3.00	0.43	11.49	0.67	14.75	0.00	0.00	6.00	0.59	0.23	0.00	4.25	60.38
2029-09	0.00	7.04	0.43	15.43	7.00	1.17	21.33	1.79	32.05	0.00	0.01	13.00	0.87	1.35	0.57	7.95	109.98
2029-10	0.16	3.56	0.00	12.89	4.00	2.11	9.39	1.27	16.70	0.00	0.11	8.00	0.95	0.71	0.06	3.79	63.71
2029-11	0.53	4.83	0.67	21.02	4.00	1.73	14.50	0.89	18.77	0.00	0.12	11.00	0.46	0.62	0.00	5.91	85.04
2029-12	0.33	3.85	0.36	16.10	4.00	2.83	10.60	0.76	17.72	0.00	0.47	10.00	1.04	0.52	0.00	5.00	73.59
2030-01	0.78	6.98	0.95	22.01	9.00	2.16	20.95	1.95	26.88	0.19	0.01	11.00	1.11	1.38	0.00	9.12	114.48
2030-02	0.00	4.11	0.39	18.35	3.00	1.79	17.60	0.81	11.68	0.00	0.01	5.00	1.97	0.59	0.39	2.73	68.41
2030-03	0.30	4.69	1.03	33.29	9.00	3.52	22.02	2.59	24.49	0.00	0.31	11.00	1.29	1.30	1.77	4.55	121.14
2030-04	0.20	5.70	1.25	71.27	22.00	3.78	51.75	2.90	33.30	0.21	0.68	10.00	4.31	1.99	3.42	6.60	219.36
2030-05	0.00	7.48	0.37	13.09	3.00	1.03	22.08	1.53	28.17	0.00	0.01	14.00	1.10	1.50	0.09	8.85	102.31
2030-06	0.16	4.13	0.37	5.39	4.00	2.37	7.93	0.51	13.37	0.26	0.48	8.00	0.52	0.45	0.00	4.62	52.55
2030-07	0.19	4.26	0.14	10.37	1.00	0.67	12.54	0.94	20.03	0.00	0.01	9.00	0.18	0.41	0.00	6.01	65.75
2030-08	0.00	5.08	0.00	6.50	1.00	1.52	9.11	0.95	16.86	0.00	0.12	3.00	0.92	0.52	0.00	3.82	49.40
2030-09	0.17	4.09	0.33	6.44	8.00	0.42	10.46	1.28	12.54	0.19	0.01	9.00	0.25	0.87	0.00	4.21	58.26
2030-10	0.00	3.86	0.19	4.11	4.00	0.85	8.79	0.91	19.12	0.00	0.00	6.00	0.39	0.59	0.00	6.02	54.82
2030-11	0.00	6.38	0.51	11.06	6.00	2.66	19.79	1.24	28.83	0.00	0.47	14.00	0.00	1.07	0.67	8.22	100.91
2030-12	0.00	1.51	0.00	3.43	4.00	1.55	1.80	0.61	7.12	0.00	0.02	3.00	0.00	0.16	0.21	0.70	24.10

Figure 3.6 presents the total expected spare part replacements per month for the entire fleet. This is the most aggregated view of the demand forecast. The plot clearly illustrates that the expected demand is highly uneven, a direct consequence of how maintenance tasks are bundled into packages in the underlying schedule. This volatility creates sharp peaks in resource needs, with some demand spikes being over four times greater than the demand in quiet months. Because demand shifts so drastically, a planner cannot rely on intuition or static reorder rules, as a strategy that works for a quiet month would likely cause a stockout during a peak. This fluctuating, non-steady demand profile is precisely what an effective procurement strategy must be able to handle.

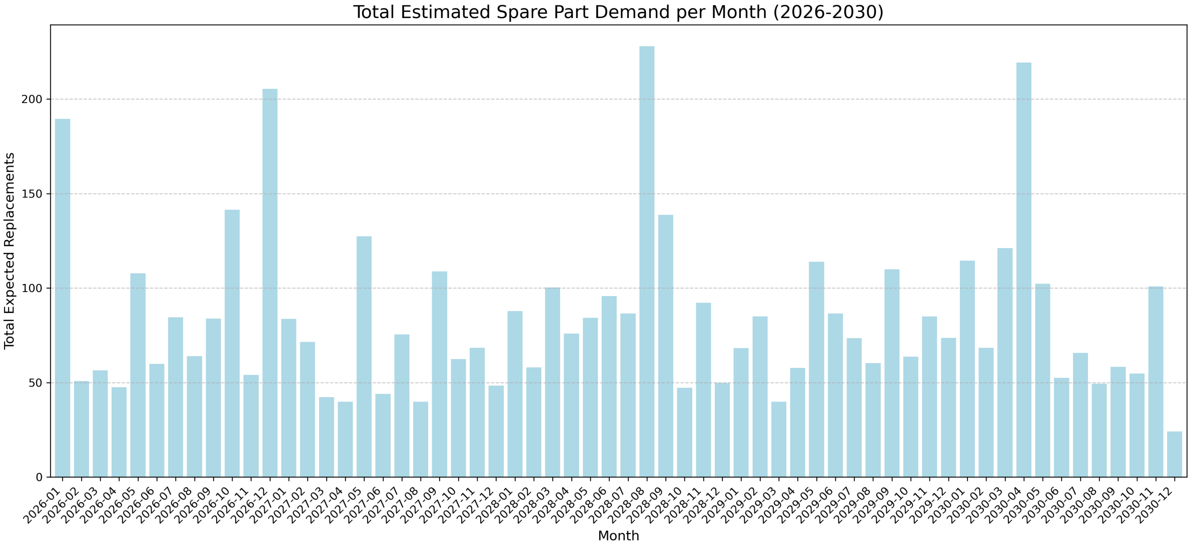


Figure 3.6: Total Estimated Spare Part Demand per Month (2026-2030)

4 Procurement & Inventory Optimization

Inventory and procurement planning for aircraft maintenance is the crucial link between a fixed schedule of needs and an unpredictable global supply chain. The core challenge, deciding what parts to order and when, forces planners to navigate significant uncertainty in demand forecasts and delivery times. This problem is magnified by a massive parts catalog and a constant stream of new data. The model presented in this chapter is built specifically for this dynamic environment. It focuses solely on the materials side, treats the maintenance schedule as a given, and provides a robust way to handle the daily flow of (uncertain) information.

This chapter introduces a model built for practicality, not theoretical perfection. This is a deliberate design choice. Rather than an all-encompassing ”optimal” solution that might be slow and opaque, the focus is on a decision-making system that is quick to recalculate, easy to explain to stakeholders, and simple to adjust. In a real-world setting, a transparent recommendation that can be generated in minutes is more valuable than a perfect one from a ”black box” that takes hours to run. Because both cost and service levels are critical, the model is designed to build trust by showing users exactly how and why specific choices are made. The following pages will detail the model and its principles, aiming to provide a clear and reliable foundation for making decisions in this complex environment.

4.1 Problem Definition & Core Principles

This section establishes the foundation for the proposed model by defining the problem from an operational perspective. The aim is to explain what a solution must accomplish and why a tailored approach is necessary. While many powerful inventory models exist, they are often built for more general applications or include a level of detail that makes them ill-suited for the specific challenges of this environment. The unique context of planned aircraft maintenance, with its calendar-driven demand and specific sources of uncertainty, calls for a solution built explicitly for this purpose.

The discussion begins by outlining the practical Requirements of a workable system, focusing on transparency, speed, and stability. It then explores the Unique Properties of the procurement environment that shape the model's design. To provide context, a brief review of Common Models from the literature highlights why their structures are not a natural fit. The section concludes with a formal Problem Summary that condenses these principles into a clear set of objectives. Together, these elements provide the justification for the model developed in the rest of the chapter.

Requirements

For this model to be useful in a daily operational setting, its design is guided by a few core principles. The approach must be **fast**, meaning it needs to process thousands of parts quickly enough for a planner to test different 'what-if' scenarios and allow for rapid recalculations after a schedule change (although this dynamic capability is not explicitly tested within this thesis). The model must also be **stable**, meaning its recommendations should not deviate unreasonably in response to minor data shifts. This ensures that results from different runs remain comparable. Above all, its results must be **explainable**. This requires **controllable** behavior, where the link between inputs, the model's decisions, and the final outputs is clear. This level of transparency is what makes such a model a useful decision support system, prioritizing a clear logic over fragile claims of optimality.

Decisions are driven by a **transparent** balance of costs. The model does not hide behind assumed perfect values for financial drivers like holding rates or shortage penalties, which are often uncertain. Instead, it treats them as visible levers. Planners can easily adjust the weights for holding, penalty, and expediting costs to reflect shifting priorities. This makes it straightforward to see how different financial policies affect procurement decisions, turning the model into a tool for exploring tradeoffs.

The system must also be **adaptable**. The core logic must be able to function even as core inputs evolve, such as changes to the parts catalog or updates to supplier lead times or other operational data. Crucially, it allows for part-level **configuration**. A highly critical component with a risky supply chain requires different treatment than a common part with steady demand, and the model accommodates these nuances through simple input adjustments, not by modifying the core method.

Finally, every recommendation must be fully **traceable**. The system records the exact state of information that led to a specific decision, including stock levels, open orders, and demand forecasts. This audit trail is essential not only for handling unexpected events in a controlled way but also for building trust. Planners and approvers can quickly verify or challenge the logic behind any action without a complex investigation.

Unique Properties

The model's unique properties are a direct response to the specific needs of an operational environment. First, unlike generic inventory systems, this policy is anchored to the maintenance calendar. It converts maintenance tasks into a rolling, probabilistic demand signal and makes decisions on the same daily timeline that planners use. This framing is necessary because the reliability of advance demand information improves as a maintenance task approaches, so the logic must update day by day. This calendar aligned, rolling approach is supported by recent work from Zhu et al. [2020], which shows the value of using planned maintenance as a driver for inventory decisions.

The decision environment also involves two major sources of uncertainty. First, demand is intermittent and driven by stochastic failure processes that are beyond the scope of this thesis, requiring the model to function well even with imperfect inputs. Second, procurement lead times are variable. By allowing for expediting, the model treats supply risk as a choice, not just an external factor. This design aligns with research on robust spare parts management, which uses expediting as a controlled lever to manage risk [Kang et al., 2023]. The resulting rule set is therefore not a generic reorder point, but a calibrated set of actions that transparently prices risk and can adjust the supply lead time when it is the optimal decision.

Finally, the approach is designed with operational realities in mind. Since inputs are often incomplete or noisy, the model treats crucial variables like failure probabilities and lead time profiles as visible and editable parameters, not hidden constants. Furthermore, the framework is configurable at the item level. This allows for tailored handling because, as research in multicriteria classification shows, spare parts differ in their criticality and supply volatility and therefore require differentiated management [Roda et al., 2014]. A detailed audit trail then records the specific conditions that justified each order, providing a clear link between the data and the outcome.

Common Models

Before introducing the deterministic model, a short tour of standard approaches for sequential decisions is helpful: dynamic programming, mixed integer programming, multistage stochastic programming, and deep reinforcement learning. The aim is context, not a survey. These methods are well established in inventory and maintenance, yet they carry limits that matter here, including rapid growth of the state with many parts, rigid time grids, and heavy training or data needs. This brief review motivates the choice of a daily rule with explicit trade offs and traceable costs.

Dynamic Programming is a strong tool for sequential decisions and can work well on compact, stationary problems. In this setting a single item still carries a rich state: calendar-dated demand tied to maintenance events, uncertain lead times that must be tracked by their residual distributions, an express option with certain arrival time, and the possibility to sell. Capturing these elements in a DP requires a detailed state description over the calendar and a discretization of the distributions, which leads to very large value tables even per item [Tiemessen et al., 2017]. In addition, the solution would need to be rebuilt frequently as the plan and supplier information change, and it must remain easy to audit and tune. Given these turnaround and transparency requirements, an exact DP is not a practical choice here. This combination of high computational cost and the opaque "black box" nature of a large value table makes it a poor fit for a system that must be fast, auditable, and easy to control.

Linear programming and Mixed Integer Linear Programming are standard tools for optimization but are ill suited for the dynamic, non-linear environment of this problem. A standard MILP formulation requires a fixed, discretized time horizon [Obradovic, 2023], which conflicts with the fluid, rolling nature of the maintenance calendar. More importantly, the core logic of the proposed deterministic model is non-linear. The cost function includes probabilistic outcomes and step function penalties, which cannot be accurately represented in a linear model without significant simplifications. These simplifications would compromise the transparency required for risk and cost traceability. The model's daily, expected value logic is a more natural fit, providing a computationally fast and flexible solution that adapts to a multi-echelon problem without the complexity of a large scale MILP [Zhang et al., 2021].

Multi-stage linear programming and its variants, such as stochastic programming, are prominent in the literature for modeling sequential decisions under uncertainty. A key example is the two-stage stochastic programming model proposed by Qin et al. [2020]. This model assists MRO providers by integrating maintenance and procurement planning. The first stage involves making initial decisions, like which parts to pre-purchase and what overhaul strategies to pursue, before the exact nature of future failures is known. In the second stage, once demand scenarios become clear, the model determines any necessary emergency procurement to address shortfalls. This two-stage process aims to minimize total costs by balancing upfront spending with potential penalties from a lack of parts. Another example is the integer programming model from Erkoc and Ertogral [2016], which optimizes the scheduling of component overhauls to align with limited repair capacity and available spare parts, minimizing premature replacements. The challenge is not just the dual-level stochasticity (in demand and lead times), but that the lead time uncertainty is endogenous: it is activated only after a procurement decision is made. This means new lead time scenarios would branch off from every potential procurement decision, which itself depends on the demand scenarios. This nested dependency would cause the problem's dimensionality to explode, making it computationally intractable.

Stranieri et al. [2024] designed a hybrid for a two echelon divergent chain that lets **deep reinforcement learning** pick production quantities while a two-stage stochastic program allocates shipments across warehouses with fixed and variable logistics costs under seasonal demand. The hybrid is motivated by the scenario growth that multi stage models face and by the sensitivity and training burden of standalone deep RL. In tests it beats a PPO agent and classic reorder rules, stays near exact solutions on small instances, and keeps a clear lead as instances grow. The setting is single item across multiple warehouses with periodic review and importantly it assumes no production or transport lead times. In the context of this thesis, the maintenance calendar creates day dated needs for many items, lead times are random and can be shortened by expediting, and buyers must audit each recommendation and re run after plan changes. This hybrid approach is not a practical fit. It would require training a separate DRL "black box" for each part, which conflicts with the need for transparency and simple configurability. Furthermore, its design requires solving a new stochastic program at every decision step, a process which itself requires building a new scenario tree to represent future uncertainty. Scaling this to thousands of parts would make the rapid recalculation required for decision support computationally impractical.

Problem Summary

Now that the general challenges have been introduced, it is time to formalize the problem. This section defines main goals the procurement model must pursue and the ground rules it must obey. Together, these elements provide a clear blueprint for the model developed in the following sections.

To make the following description clear, we'll use a few simple variables. We are looking at a set of planning days D, and τ represents any single day within that set. On day τ, there is a demand for parts E_τ and a quantity of arriving parts A_τ. The inventory on hand at the end of the day is i_τ (making $i_{\tau-1}$ the inventory available at the start of the day). For costs, P is the unit purchase price and h is the daily holding cost per unit. It is important to note that this notation is simplified for conceptual clarity. It is only used to define the high-level problem here and is not the final, more detailed notation used in the model implementation later in the chapter.

Core Objectives

The model's primary task is to intelligently balance a set of competing priorities. The following table presents the main, though not exhaustive, list of objectives that guide its decisions. Each represents a standalone ambition.

Aim	Performance Measure	Target
Minimize service failures	The probability of a stockout on any day with maintenance: $\sum_{\tau \in D} \mathbb{P}\big(E_\tau > i_{\tau-1} + A_\tau\big)$	$\downarrow$ (Minimize)
Minimize time in stock	Total holding cost, which penalizes keeping inventory for too long: $\sum_{\tau=1}^{d_{\text{end}}} h \cdot \mathbb{E}[i_\tau]$	$\downarrow$ (Minimize)
Spend prudently	The expected total purchasing spend over the horizon: $\sum_{\tau=1}^{d_{\text{end}}} P \cdot \mathbb{E}[\text{orders placed on day } \tau]$	$\downarrow$ (Minimize)

The goal to minimize service failures is arguably the most important. In this business, a stockout is not a minor inconvenience. It can mean a maintenance check is delayed and an aircraft is grounded, which creates huge knock-on costs and operational chaos. This puts a lot of pressure on having the right part at the right time. It is important to clarify that this thesis does not model the specific operational consequence for each individual part. Instead, these "huge knock-on costs" are represented by a generalized financial penalty, which is scaled based on a part's price and category (e.g., Expendable, Repairable) rather than a part-specific impact study. On the other side of the coin, we have to watch the time in stock. Aircraft parts are often very expensive, and letting millions of Euros in components sit on a shelf is a poor use of company money. This is what the holding cost in the model represents. Lastly, the need to spend prudently simply reflects the fact that there isn't an unlimited budget for parts.

These goals are always in conflict. The most obvious way to avoid grounding an aircraft is to buy plenty of every part that might ever be needed. But this approach results in spending a fortune on both the parts themselves and on the cost of storing them, completely failing at the other two goals. If the strategy swings too far the other way, attempting to save money by only ordering parts at the last minute, it will almost certainly result in stockouts and the massive penalties that come with them. The model's job is not to eliminate this conflict. Instead, its purpose is to find a sensible middle ground, using the cost settings to make a logical trade-off between buying protection and saving money.

Fundamental Constraints

Any feasible plan must operate under a set of fundamental constraints. The following list is not exhaustive but highlights the key rules that shape the problem.

1. **Information Integrity:** Decisions can only be based on information that is currently known. Beliefs about future events, such as demand and arrivals, are represented by formal probability distributions. These beliefs are updated consistently over time by conditioning only on new, realized information; past beliefs cannot be retroactively changed.

2. **Irreversible Commitments:** A procurement decision made on a given day is final. It cannot be revised at a later date, even if new information becomes available that might have altered the original choice.
3. **Immediate Cancellation for Shortages:** If demand on a given day exceeds the available supply (defined as on-hand inventory plus any new arrivals), the unmet portion is not backordered. Instead, the corresponding maintenance task is considered canceled. This incurs a fixed cancellation penalty, which is the model's sole method for translating all subsequent operational consequences (like delays or grounding) into a single financial figure. To avoid conflicting motives, the model also incurs the purchase cost of the shortfall units.
4. **Cost-Based Justification:** An action, such as placing a new order, should only be taken if the total expected future cost (including purchasing, holding, and penalties) is lower than the expected cost of simply waiting and taking no action.

This summary establishes the blueprint for the model by outlining its high-level goals and the rules it must follow. The following sections will transition from this conceptual framework to the specific implementation, detailing the mechanics of how the model operates to achieve these objectives under uncertainty.

4.2 Proposed Deterministic Model

The principles of speed, transparency, and traceability are directly built into the model's functional design. To explain how decisions are made, the architecture is presented in three progressive layers. The process begins with pre-procurement, which establishes a fixed schedule of incoming parts that acts as a baseline for the system. Building upon this, the core of the model is a single-day evaluator. This component calculates the total expected future cost of any potential action (like buying or waiting), using a detailed short-term forecast for immediate accuracy and a more general long-term forecast for strategic control. Finally, a master routine uses these cost evaluations to intelligently select the best action for each part, every day. To ensure the logic is fully transparent, the necessary notation is introduced as each component is explained.

4.2.1 Pre-procurement

In practice, not all procurement is reactive. Planners often establish standing agreements for a steady supply of materials based on long-term forecasts and contracts. To reflect this, our model first establishes a **pre-procurement schedule**: a predictable stream of supply that arrives automatically.

This can be thought of as a baseline subscription for parts. These orders are not decided by the daily decision model; they are treated as fixed, incoming deliveries with their costs already committed. This design allows the optimization algorithm to focus on its primary function: managing the uncertain gaps between this baseline supply and the actual, fluctuating demand. The model's role is to observe this baseline, monitor realized outcomes, and then make calculated, corrective actions when necessary.

Implementation

The pre-procurement schedule is determined once at the beginning of the planning horizon using a transparent logic based on two policy levers: a **confidence level (γ)** for sizing the quantity, and a **delivery interval (c)** for timing the arrivals. The goal is to secure a conservative amount of inventory that is highly likely to be used and to schedule its arrival in a steady, predictable pattern.

The process for each part m involves two steps:

1. **High-Confidence Quantity Sizing.** We first consider the total expected demand for part m over the entire horizon (d_{end} days). This is $X_m = \sum_{\tau=1}^{d_{\text{end}}} \mathbb{E}[E_{m,\tau}]$, where $\mathbb{E}[E_{m,\tau}]$ is the mean expected demand for part m on day τ. Instead of using the mean expected demand, we select a quantity Q_m^{pre} that is highly likely to be consumed. This value represents the largest quantity that is expected to be consumed with at least a probability of γ. This method minimizes the risk of procuring inventory that results in excess stock. Formally, it is the lower quantile of total demand:

   $$Q_m^{\text{pre}}(\gamma) = \max\{\, q \in \mathbb{Z}_{\geq 0} : \Pr[X_m \geq q] \geq \gamma \,\}$$

2. **The Delivery Schedule.** To avoid a single large influx of inventory, the total quantity Q_m^{pre} is distributed across K delivery dates. These dates are determined by the interval c, with $K = \lfloor d_{\text{end}}/c \rfloor$ deliveries planned evenly over the horizon. Let the average quantity per delivery be $a = Q_m^{\text{pre}}/K$. The allocation method depends on this average.

   - *If $a \geq 1$ (Sufficient Totals):* When the total quantity is large enough to supply at least one unit per delivery date, we assign equal drops. Any remainder is intentionally ignored for simplicity.

     $$q_{\text{drop}} = \lfloor Q_m^{\text{pre}}/K \rfloor$$

   - *If $a < 1$ (Sparse Totals):* When the total quantity is smaller than the number of delivery dates, a simple division would result in zero for most drops. To allocate the few units as evenly as possible, we use a cumulative rounding scheme. For each delivery date $j = 1, \ldots, K$, the quantity to schedule, s_j, is calculated as:

     $$s_j = \Big\lfloor (1-a) + ja \Big\rfloor - \Big\lfloor (1-a) + (j-1)a \Big\rfloor$$

     This standard technique ensures that exactly Q_m^{pre} units are scheduled in single-unit increments, distributed smoothly across the K available dates.

These scheduled deliveries become fixed inputs for the daily optimization. While in operational settings pre-procurement might cover a substantial portion of total demand, for this study we deliberately select a high confidence level γ. This choice results in a smaller, more conservative baseline of supply, intentionally leaving a larger share of the demand uncertain. The purpose is to create a more challenging environment to rigorously test the daily model's decision-making capabilities. This approach provides a stable but minimal foundation, requiring the optimization to actively manage the majority of the day-to-day uncertainty.

4.2.2 Deterministic Model

Before presenting the formal equations, it is useful to describe the model in plain language. The goal is to provide a clear conceptual understanding of the model's logic so a reader can anticipate its behavior. The mathematical formulation that follows formalizes these concepts, detailing exactly how expectations, costs, and decisions are computed.

Each day, the model observes the current state of inventory and the latest information on future events. It then selects an action described by purchase/selling quantity. A third option is an express order placed the day before need, arriving the next day at a high add-on cost set below the shortage penalty. This action is autonomously activated on that day whenever the expected penalty from insufficient stock exceeds the express cost, so the order is placed regardless of other decisions or longer term considerations. Decisions are based on financial factors, including costs for cancelling maintenance, holding costs and purchase costs. To make the best choice, the model forecasts the expected future that would result from each candidate action and compares their associated costs. The next day begins with a new state, which reflects what actually materialized from the previous day, including new arrivals and realized demand. This design allows the policy to be highly reactive on a daily basis while still utilizing a forward looking perspective that is recalculated whenever new information arrives.

The ability to sell is a key feature that makes this model unique. While most procurement models treat stock as a one-way sink, many spare parts can be sold at a meaningful price, especially when other organizations face shortages. In this model, selling is not a profit driver but a safety valve that makes buying decisions more flexible in volatile periods. The model can choose to sell if projected inventory comfortably exceeds future needs and the expected reduction in holding and penalty costs outweighs the lost units. This keeps the focus on overall service and total cost rather than speculative gains.

The implementation simulates decisions over the full planning horizon to report aggregate performance, but its operational rhythm is daily. Inputs on any given day depend on the decision made the day before and on any arrivals or demand observed overnight. This rolling structure is vital for two reasons. First, it aligns with how planners work and how data naturally becomes available. Second, it ensures the model can be recomputed quickly after any schedule update, supplier delay, or change in demand forecasts.

For regular orders, lead times are uncertain but their distributions are assumed to be known. This leads to a split of the planning horizon into a short term and a long term for each decision day. The short term is defined as the period in which a newly ordered unit could still arrive. The policy evaluates all relevant scenarios exactly in this short-term window, where timing is critical. It then uses moment summaries to control overstocking and risk in the long term, avoiding unnecessary computation. The cost signal guiding decisions combines purchase, holding, and penalty terms, and is controlled by tunable policy weights. The goal is to make these trade-offs explicit so that a change in policy or data produces a predictable and explainable change in behavior.

The model also carries limitations that stem from scope and design rather than intent. It does not learn during execution, so decisions reflect the input prices, penalties, holding rates, selling fees, and lead time beliefs exactly as provided. Results should therefore be interpreted as policy outcomes under stated assumptions, not as claims of optimality under the true environment. The emphasis on a short term window reduces computation but can tilt choices toward immediate savings and understate rare severe shortages. Influence on long term service rates and capacity is indirect and operates through policy weights, budget rules, and the express mechanism. Mitigations in this chapter are pragmatic rather than algorithmic learning: record the assumptions that drive each decision, apply stress and sensitivity checks, and use simple guardrails to prevent extreme recommendations.

Notation

The notation used throughout this chapter is presented here. The variables and sets are categorized to distinguish between high level concepts and the specific parameters that track the model's state. Decision variables represent the actions the model can take each day, such as placing a regular or express order, or selling an item. Tracking variables are used to capture the model's internal state and evolving beliefs about the future, including on hand inventory and the probabilistic forecasts of future arrivals. For detailed definitions of the data structures that underpin the model's state, such as the open orders set $A_{m,d}^{\text{expected}}$ and the inventory forecast distribution $I_{d,\tau}^{m,\text{fore}}$, please refer to Table B.1 in Appendix B.

Sets

M	set of all maintenance tasks
$\mathcal{K}_\tau$	index set of scenario nodes at future day τ
$\mathcal{F}_\tau$	scenario set at day τ; master set $\mathcal{F} = \bigcup_\tau \mathcal{F}_\tau$
$\mathcal{E}_\tau^m = (e, p_{m,\tau}^E(e))$	Expected Demand where e is replacement quantity (check table B.1)

Decision variables

$x^{\text{reg}}_{m,d}$	regular order quantity for task m placed on day d (stochastic lead time)
$x^{\text{expr}}_{m,d}$	express order quantity for task m placed on day d (deterministic arrival at $d{+}1$)
$x^{\text{sell}}_{m,d}$	quantity sold on day d

Tracking variables

i^m_d	on-hand inventory at *end* of day d
$A^{\text{expected}}_{m,d}$	set of *per-order* expected-arrival atoms open at day d (check table B.1)
$\alpha^{\text{expected}}_{m,l,\tau}$	atom for an order of size q^m_l placed on day l with $\pi^{(d)}_{m,l}(\tau),\ \tau > d,\ \ \sum_{\tau>d} \pi^{(d)}_{m,l}(\tau) = 1$
$\alpha^{\text{reg}}_{m,l,\tau}$	realized regular arrival quantity on day τ from order day l (observation)
$\alpha^{\text{expr}}_{m,\tau-1,\tau}$	realized express arrival on day τ; equals $x^{\text{expr}}_{m,\tau-1}$ (deterministic after order is placed)
$I^{m,\text{fore}}_{d,\tau}$	forecast distribution of inventory for task m at day τ given information at day d (check table B.1)
$\mathbb{E}[I^m_\tau],\ \text{Var}(I^m_\tau)$	corresponding moments of $I^{m,\text{fore}}_{d,\tau}$

4.2.3 Input Values

Part Prices

Since component price data for the simulated A320 fleet was not available, a realistic price list for the 1,134 parts was synthesized using a real-world dataset from a different aircraft, the DHC8. The goal of this approach was not to achieve perfect price accuracy for any specific part, but rather to replicate the overall statistical properties of an operational inventory. Most importantly, this method captures the characteristic skew where a small number of high-value components account for a large portion of the total inventory value. The process involved four distinct stages:

Reference Scaling: The source DHC8 price list was uniformly scaled by a factor ($\kappa = 3.0$) to adjust its magnitude to the A320 context. This step increases the prices while perfectly preserving the shape of the original distribution.

Weighted Bootstrap: The final price list was generated by drawing with replacement from this scaled reference list. The draw was weighted using beta parameters ($\beta_{\text{EXP}} = 0.5, \beta_{\text{REP}} = 1.0, \beta_{\text{ROT}} = 2.0$) to slightly favor the selection of more expensive parts within each category, reflecting the operational importance of high-value components.

Probabilistic Soft Split: Each generated price was then categorized as Expendable (EXP), Repairable (REP), or Rotable (ROT). Instead of using soft price cutoffs, this was done via a probabilistic soft split, where parts near a boundary have a chance of falling into either category, creating a more natural classification.

Total Value Calibration: As a final step, a single multiplicative factor was applied to the entire new list. This ensures the total monetary value of all simulated parts is an exact match to the total value of the scaled reference dataset, grounding the simulation's economy to the source data.

The resulting distribution of the 1,134 synthesized part prices is shown in Figure 4.1.

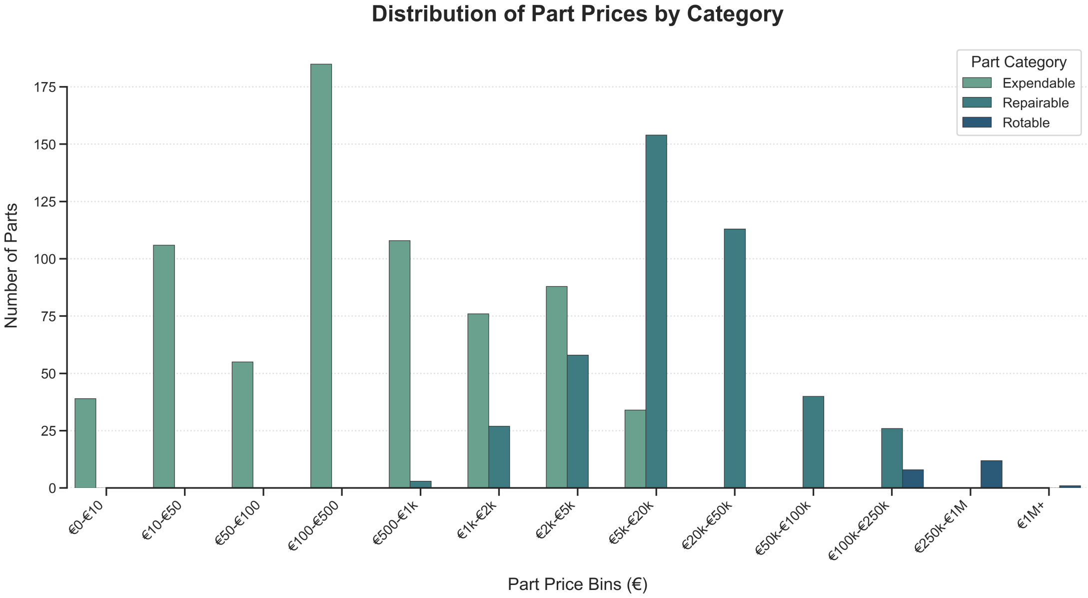


Figure 4.1: Distribution of Synthesized Part Prices.

Cost Structure

The model's economic decisions are guided by a set of cost parameters that are dynamically derived from each part's purchase price, P. This design ensures that the economic trade-offs for a 10 € part are appropriately scaled compared to those for a 100,000 € part, creating a consistent logic across the entire parts catalog.

It is important to clarify the nature of these values. The specific multipliers and base costs presented here are not derived from a formal industry study; rather, they represent the baseline configuration for the experiments conducted in this thesis. A primary objective of this work is to analyze the model's sensitivity to these economic levers, treating them as adjustable inputs rather than fixed truths. Therefore, the reader can assume these default values are in effect for all subsequent analyses, unless explicitly stated otherwise. A full sensitivity analysis on these parameters is to be presented in Chapter 5.

The primary cost relationships are defined as follows:

Shipping Costs ($C_{\text{reg}}, C_{\text{expr}}$): Both regular and express shipping costs include a small fixed fee (in Euros) plus a variable component proportional to the part's price. The express option is priced significantly higher to reflect its premium nature. Shipping costs for pre-procurement orders are assumed to be zero, as these larger, planned shipments would likely benefit from economies of scale and be handled under separate freight agreements.

- Regular Shipping: $C_{\text{reg}} = 5 + 0.01 \cdot P$
- Express Shipping: $C_{\text{expr}} = 50 + 0.10 \cdot P$
- Pre-procurement Shipping: no fee

Holding Cost (h): The cost to hold an item in inventory is based on a **10% annual rate** of the part's purchase price (P). This annual cost is converted to a daily charge, as the model operates on a daily timeline: $h = (0.10 \cdot P)/365$.

Selling Price (S): When the model sells a surplus part, it recoups **80%** of the original purchase price ($S = 0.80 \cdot P$). This value is set below the purchase price to prevent wasteful buy-and-sell loops. At the same time, it is a significant value, which gives the model flexibility. It can proactively purchase a part to cover a potential stockout, knowing that if the part is not ultimately needed, a large portion of its value can be recovered.

Cancellation Penalty (PEN^m): As noted in the 'Fundamental Constraints', a stockout incurs a fixed cancellation penalty. For this baseline configuration, this penalty is defined as a large fixed amount plus a multiple of the part's price: $\text{PEN}^m = 200,000 + 5 \cdot P$. This structure ensures that the penalty for high-value parts is appropriately scaled, while the large base cost represents the significant operational disruption of any cancellation.

Lead Time Distributions

In this thesis, lead times for regular orders are not fixed values but are treated as stochastic, reflecting the inherent uncertainty of real-world supply chains. Each part is assigned a lead time profile based on its primary category. This is achieved by modeling each lead time as a discrete **triangular distribution**.

This distribution is defined by its mode (the most likely arrival day) and its support (the range of possible arrival days). The probability is highest at the mode and decreases linearly to the edges of the support. The specific parameters for each part's distribution are determined as follows:

The Support (Range of Possible Days): The length of the potential arrival window is determined by the part's primary category. The default lengths of the support are:

- **Expendable (EXP):** 5 days
- **Repairable (REP):** 7 days

The Mode (Most Likely Arrival Day): The peak of the distribution, or most likely arrival day, is also a fixed value based on the part's category:

- **Expendable (EXP):** Day 5
- **Repairable (REP):** Day 10

For instance, consider an **Expendable** (EXP) part. Its lead time distribution has a support of 5 days and a mode set at day 5. This combination produces the following discrete triangular distribution for the arrival day offset: Day 3 (7.7%), Day 4 (23.1%), Day 5 (38.5%), Day 6 (23.1%), Day 7 (7.7%).

Guards for Selling Decisions

One of the model's unique features is its ability to sell excess inventory. However, this action requires careful control. The model's daily cost evaluation is forward-looking, meaning the purchase cost of an item already in stock is a sunk cost and is not considered in today's decision. Without safeguards, the model might be tempted to sell an item to save on a few days of holding costs, only to buy it back again later under a different cost objective. Since the selling price is lower than the purchase price ($S < P$), this loop would consistently lose money.

To prevent this costly buy-and-sell loop, the model uses a two-layer system of 'sale guards'. These guards function as a safety check, only permitting a sale when the model is highly confident that an item is truly surplus and will not be needed again soon. This system consists of a simple value threshold, followed by a sophisticated risk assessment that mathematically measures whether this confidence threshold has been met.

1. **Minimum Price Threshold:** A sale is only ever considered for parts with a purchase price of at least **500 €**. This practical rule prevents the model from wasting computational effort and operational attention on liquidating low-value stock where the transaction overhead might outweigh the benefits.
2. **Risk-Based Uncertainty Guards:** If a part is expensive enough to be considered for sale, it must then pass a risk assessment. The model projects the inventory distribution at the end of the horizon, described by its mean μ_T and variance σ_T^2. A sale that would leave q units remaining is only permitted if one of the following guards, with **Cantelli's inequality** being the default, gives the green light. Each guard offers a different way to define "sufficient certainty."

- **Cantelli's Inequality Guard:** This is the most conservative, distribution-free method. It makes no assumptions about the shape of the long-term inventory distribution. It calculates a worst-case upper bound on the probability of the inventory dropping to or below the proposed post-sale level q. The sale is only allowed if this worst-case probability is below a risk tolerance α, which defaults to **5%**.

  $$\text{Allow sale if:} \quad \frac{\sigma_T^2}{\sigma_T^2 + (\mu_T - q)^2} \leq \alpha$$

- **Normal Approximation Guard:** This guard assumes the long-term inventory distribution is approximately Normal, $\mathcal{N}(\mu_T, \sigma_T^2)$. It calculates a safety stock level based on the desired service level $(1-\alpha)$. A sale is only allowed if it does not consume this safety stock. Here, $z_{1-\alpha}$ is the $(1-\alpha)$-quantile of the standard normal distribution.

  $$\text{Allow sale if:} \quad q \geq \mu_T - z_{1-\alpha}\sigma_T$$

- **Coefficient of Variation (CV) Guard:** This guard focuses on the stability of the forecast itself. It measures the relative uncertainty of the inventory projection over the entire long-term tail. A sale is only allowed if this forecast is deemed reliable, i.e., if its coefficient of variation is below a set threshold, z_{thresh}, which defaults to **0.10**.

  $$\text{Allow sale if:} \quad CV_{\text{tail}} = \frac{\sqrt{\text{Var}(I_{d_{\text{end}}}) - \text{Var}(I_T)}}{\mathbb{E}[I_T]} \leq z_{\text{thresh}}$$

Each of these guards provides a different mathematical approach to achieve the same goal: to ensure that selling is a deliberate strategic action, taken only when there is a high degree of confidence that the stock is truly surplus. For the purposes of this thesis, the Cantelli's inequality guard is used as the default. This choice is based on initial testing: holding all other model parameters constant, the Cantelli metric empirically resulted in the lowest final net cost for the simulation when compared to the other methods.

Part Categories

Expendable Parts represent the simplest case. When a task requires an expendable part, it is consumed and removed from service permanently. Therefore, any demand for an expendable part always translates into a requirement for one new unit from stock.

Repairable Parts follow a simplified repair loop. When a maintenance task identifies a faulty repairable part, two main outcomes are possible:

- **Scrapped (25% chance):** The part is Beyond Economic Repair. This creates an immediate demand for one new replacement unit from stock.
- **Repairable (75% chance):** The part can be repaired. This path is then split:
  - **On-Spot Repair:** (1/3 of repairable cases) The part is fixed on the aircraft, creating no demand for a stock unit.
  - **Swap & Repair:** (2/3 of repairable cases) The part must be sent away and is swapped with a serviceable unit from inventory. This swap creates a demand for one unit.

If a swap is required but no unit is in stock, the maintenance task is considered cancelled and incurs a penalty. Any part sent for repair is assumed to return to inventory in "like-new" condition after either one or two days, with equal probability.

Rotable Parts were excluded from this study entirely. This decision stems from the fundamentally different logistics model governing these parts. Rotables are typically not procured on a per-demand basis but are managed through ”pools” where unserviceable units are swapped for serviceable ones from a circulating stock. Furthermore, their characteristically high cost presents a significant challenge for model validation. Including them would disproportionately skew the aggregate financial metrics. To ensure the evaluation accurately reflects the model’s effectiveness on the bulk of the inventory, rotable parts were deliberately scoped out of the analysis.

Algorithm

Before introducing the master routine that selects the daily action, we first explain the evaluation process for a single candidate action. Algorithm 3 is the core engine that computes the total expected future cost of a candidate decision (e.g., buy, sell, or wait) made on day d for item m. To balance detailed accuracy in cost expectations with computational tractability, the evaluation horizon is split into two parts. The short-term window $\mathcal{T}$ extends to day $T = \min\{d + L_{\text{max}}, d_{\text{end}}\}$, where L_{max} is the maximum lead time, while the long-term tail $\mathcal{T}_{\text{LT}}$ covers the remaining horizon. This dual-horizon approach is a deliberate trade-off: it provides a high-fidelity analysis for near-term events where timing is critical, while maintaining a computationally efficient, strategic view of the distant future.

In the short-term window, future states are projected using full probability distributions, while in the long-term tail, outcomes are approximated using moment propagation. For the short term, the model performs a detailed scenario analysis by explicitly mapping out every possible combination of events, such as different demand quantities and all potential arrival dates for open orders. Each complete scenario has a specific, known probability of occurring, allowing the model to calculate the total expected cost as a weighted average of all these outcomes. For the long-term “tail,” this full tracking becomes too computationally expensive. The model instead uses a more efficient approximation: it takes the final expected inventory level and its variance from the end of the short-term window and projects these two values forward, day by day, by incorporating the mean and variance of future demand, also accounting for pre-procurement arrivals.

A key component of the cost function is the expected shortfall cost, which is calculated using the function $\Psi(q, e)$. This function computes the cost for any day by coupling the full probability mass functions (pmfs) of demand and inventory. Its formulation captures both the operational disruption (the fixed penalty PEN^m when demand e exceeds inventory q) and the immediate financial impact of sourcing a replacement under pressure (the variable cost P^m per missing unit). The function is defined as $\Psi(q, e) = \text{PEN}^m \cdot \mathbf{1}\{e > q\} + P^m \cdot \max\{e - q, 0\}$.

The evaluation begins with state transitions. The algorithm advances the system from day $d - 1$ to d by applying any realized arrivals to the on-hand inventory, i^m_d, and updating the set of open orders, $A^{\text{expected}}_{m,d}$. It then applies the candidate action to this new state, for example by reducing i^m_d for a sale or adding a new regular order to $A^{\text{expected}}_{m,d}$. With this updated state, it constructs the complete scenario space over the short-term window $\mathcal{T}$, enumerating all possible combinations of arrival times for open orders and demand realizations. This scenario enumeration produces inventory PMFs $\{\mathcal{I}^m_\tau\}$ for each day in the window, derived from a full expansion of all possible outcomes.

The express order mechanism functions as a critical fallback, intended to prevent imminent stockouts only when regular and pre-procurement actions have proven insufficient. This option is only evaluated on the day immediately preceding a maintenance task ($d = d^m_j - 1$) for a simple mechanical reason: it has a deterministic, one-day arrival time. Even in this reactive mode, the decision is still cost-based. The model runs a local optimization, comparing the high cost of an express order against the expected penalty of a stockout on the following day. An order is placed only if it is the more economical choice at that specific moment.

Finally, the total expected cost V^* is assembled. Short-term costs for holding, purchasing, and penalties are calculated over the window $\mathcal{T}$, using the Ψ function to compute shortfall costs on all relevant maintenance days. For the long-term period $\mathcal{T}_{\text{LT}}$, where full enumeration is computationally infeasible, the model propagates the first two moments of the inventory distribution. It identifies potential excess inventory and schedules a projected sale on the earliest future day that satisfies a risk-based guard based on Cantelli's inequality. This guard ensures that a sale is only considered when the inventory forecast is stable, preventing the premature disposal of stock that might be needed later to cover unexpected demand. Long-term holding costs are then adjusted to reflect this planned sale. Total value V is the sum of all costs less any revenue from an immediate or projected sale. This final cost, along with any auto-generated express order and the resulting state, is returned to the master routine to facilitate a fair comparison between all candidate actions.

Algorithm 3. **Expected Cost Evaluation for Procurement Decision $(x^{\text{reg}}, x^{\text{sell}})$ on Day d**

```
Input: Part $m$; Current day $d$; State: $i^m_{d-1}$ (inventory), $A^{\text{expected}}_{m,d-1} = \{(q_\ell, \pi_\ell, \ell) : \ell < d\}$ (open orders with
       quantity, PMF, order day);
Action: $x^{\text{reg}}$ (regular order), $x^{\text{sell}}$ (sell quantity); Data: $\{\mathcal{E}^m_\tau\}_{\tau \geq d}$ (demand PMFs), $\{d^m_j\}_{j \in J^m}$ (maint. days),
$\Pi^{\text{LT}}_m$ (lead time PMF), $\{\alpha^{\text{pre}}_{m,\tau}\}_\tau$ (pre-procurement); Costs: $P^m$, $h_m$, $S^m$, $\text{PEN}^m$, $C^m_{\text{reg}}$, $C^m_{\text{expr}}$; Risk tolerance: $\alpha$
Output: $V^*$ (total expected cost), $x^{\text{expr}}$ (express order), $A^{\text{expected}}_{m,d}$ (updated orders), $\{\mathcal{I}^m_\tau\}_{\tau \in \mathcal{T}}$ (inv. PMFs)

// State transition from $d-1$ to $d$
$i^m_d \leftarrow i^m_{d-1} - x^{\text{sell}} + \alpha^{\text{pre}}_{m,d} + \sum_{\ell<d} q_\ell \cdot \mathbf{1}\{\text{arrival realized on } d\}$    // Apply action & arrivals
$A^{\text{expected}}_{m,d} \leftarrow \emptyset$    // Rebuild open orders
for $(q_\ell, \pi_\ell, \ell) \in A^{\text{expected}}_{m,d-1}$ do
    if $\sum_{\tau>d} \pi_\ell(\tau) > 0$ then
        $\pi'_\ell(\tau) \leftarrow \pi_\ell(\tau) / \sum_{s>d} \pi_\ell(s)$ for all $\tau > d$    // Condition on not arrived yet
        $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \cup \{(q_\ell, \pi'_\ell, \ell)\}$
if $x^{reg} > 0$ then
    $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \cup \{(x^{\text{reg}}, \Pi^{\text{LT}}_m, d)\}$    // Add new order
// Define evaluation windows
$\mathcal{T} \leftarrow [d+1, \min\{d + \max_\tau\{\tau : \Pi^{\text{LT}}_m(\tau) > 0\}, d_{\text{end}}\}]$    // Short-term: possible arrival window
$\mathcal{T}_{\text{LT}} \leftarrow [\max(\mathcal{T}) + 1, d_{\text{end}}]$    // Long-term: beyond arrival window
// Construct scenario space over $\mathcal{T}$ (See alg. 5 and alg. 6)
$\mathcal{A} \leftarrow \{a : A^{\text{expected}}_{m,d} \times \mathcal{T} \to \{0,1\}\}$ where $a_{\ell,\tau} = 1$ iff order $\ell$ arrives on day $\tau$
$\mathcal{A}_{\text{valid}} \leftarrow \{a \in \mathcal{A} : \sum_\tau a_{\ell,\tau} \leq 1 \text{ for each } \ell\}$    // Each order arrives at most once
$\mathcal{D} \leftarrow \prod_{\tau \in \mathcal{T}} \text{support}(\mathcal{E}^m_\tau)$; $\mathcal{S} \leftarrow \mathcal{A}_{\text{valid}} \times \mathcal{D}$    // All demand realizations & Complete scenario set
// Compute inventory PMF at each $\tau \in \mathcal{T}$ via scenario aggregation
for $\tau \in \mathcal{T}$ do
    $\mathcal{I}^m_\tau \leftarrow \{\}$    // Initialize as empty PMF
    for $(a, e) \in \mathcal{S}$ do
        $Q^{(a)}_\tau \leftarrow \sum_{(\ell,t) \leq \tau} a_{\ell,t} \cdot q_\ell$    // Total arrivals by $\tau$ in scenario $a$
        $D^{(e)}_\tau \leftarrow \sum_{t \leq \tau} e_t$    // Total demand by $\tau$ in scenario $e$
        $I^{(a,e)}_\tau \leftarrow i^m_d + Q^{(a)}_\tau + \sum_{t \leq \tau} \alpha^{\text{pre}}_{m,t} - D^{(e)}_\tau$    // Inventory level
        $p^{(a,e)} \leftarrow \left[\prod_{\ell,t} \pi_\ell(t)^{a_{\ell,t}} \cdot (1 - \sum_{s \leq \tau} \pi_\ell(s))^{1 - \sum_s a_{\ell,s}}\right] \cdot \left[\prod_t p^{\mathcal{E}}_t(e_t)\right]$
        $\mathcal{I}^m_\tau[I^{(a,e)}_\tau] \leftarrow \mathcal{I}^m_\tau[I^{(a,e)}_\tau] + p^{(a,e)}$    // Aggregate probabilities by inventory level
    $\mu_\tau \leftarrow \sum_{(q,p) \in \mathcal{I}^m_\tau} q \cdot p$;   $\sigma^2_\tau \leftarrow \sum_{(q,p)} (q - \mu_\tau)^2 \cdot p$    // Compute moments
// Express order optimization (activated only if $d = d^m_j - 1$ for some $j$)
$x^{\text{expr}} \leftarrow 0$    // Default: no express
if $\exists j \in J^m : d = d^m_j - 1$ then
    Define $\Psi(q, e) \equiv \text{PEN}^m \cdot \mathbf{1}\{e > q\} + P^m \cdot \max\{e - q, 0\}$    // Shortfall cost function
    Define $\mathcal{C}_{\text{expr}}(x) \equiv x(P^m + C^m_{\text{expr}}) + \sum_{(q,p_q) \in \mathcal{I}^m_{d+1}} \sum_{(e,p_e) \in \mathcal{E}^m_{d+1}} p_q p_e \cdot \Psi(q + x, e)$
    $x^{\text{expr}} \leftarrow \arg\min_{x \in \{0,1,\ldots,\bar{x}_{\max}\}} \mathcal{C}_{\text{expr}}(x)$    // Minimize expected cost
    if $x^{expr} > 0$ then
        for $\tau > d$ do
            $\mathcal{I}^m_\tau \leftarrow \{(q + x^{\text{expr}}, p) : (q, p) \in \mathcal{I}^m_\tau\}$    // Add express qty to all inventory levels
// Short-term cost accumulation over $\mathcal{T}$
$V_{\text{hold}} \leftarrow \sum_{\tau \in \mathcal{T}} h_m \cdot \mu_\tau$    // Expected holding cost
$V_{\text{pur}} \leftarrow x^{\text{reg}}(P^m + C^m_{\text{reg}}) + x^{\text{expr}}(P^m + C^m_{\text{expr}})$    // Purchase & shipping
$V_{\text{pen}} \leftarrow \sum_{j : d^m_j \in [d, \max(\mathcal{T})]} \sum_{(q,p_q) \in \mathcal{I}^m_{d^m_j}} \sum_{(e,p_e) \in \mathcal{E}^m_{d^m_j}} p_q p_e \cdot \Psi(q, e)$    // Expected penalties
// Long-term analysis via moment propagation over $\mathcal{T}_{LT}$
$(\mu_t, \sigma^2_t) \leftarrow (\mu_{\max(\mathcal{T})}, \sigma^2_{\max(\mathcal{T})})$    // Initialize from end of short-term
for $t \in \mathcal{T}_{LT}$ do
    $\mu_t \leftarrow \mu_{t-1} + \alpha^{\text{pre}}_{m,t} - \mathbb{E}[\mathcal{E}^m_t]$;   $\sigma^2_t \leftarrow \sigma^2_{t-1} + \text{Var}[\mathcal{E}^m_t]$
// Projected sale using Cantelli's inequality
$q^* \leftarrow \lfloor \min_{t \in \mathcal{T}_{\text{LT}}} \mu_t \rfloor$;   $t^* \leftarrow \infty$    // Identify stable excess
if $q^* > 0$ and $P^m \geq 500$ then
    $t^* \leftarrow \min\{t \in \mathcal{T}_{\text{LT}} : \frac{\sigma^2_t}{\sigma^2_t + (\mu_t - q^*)^2} \leq \alpha\}$    // First safe potential selling day
$V_{\text{LT}} \leftarrow \sum_{t \in \mathcal{T}_{\text{LT}}}^{t^*-1} h_m \mu_t + \sum_{t^*}^{d_{\text{end}}} h_m \max\{\mu_t - q^*, 0\} - \mathbf{1}\{t^* < \infty\} q^* S^m$
// Total expected cost
$V^* \leftarrow V_{\text{hold}} + V_{\text{pur}} + V_{\text{pen}} + V_{\text{LT}} - x^{\text{sell}} \cdot S^m$
return $V^*$, $x^{expr}$, $A^{expected}_{m,d}$, $\{\mathcal{I}^m_\tau\}_{\tau \in \mathcal{T}}$
```

The master routine, presented in Algorithm 4, functions as the strategic decision-making layer of the model. Its primary role is to synthesize the detailed cost assessments from the 'ExpectedValue' function into a single, optimal action for each part on a given day. On each day d, it methodically determines whether to wait, sell, or place a regular order by comparing the total expected future cost of each potential choice. The final output is a single, committed decision: a specific quantity to purchase or sell, along with any auto-generated express order required by that decision's projected future state. This structure separates the high-level strategic choice from the tactical, reactive logic of the express order.

The algorithm proceeds chronologically through the planning horizon. For the purpose of this simulation, it processes each part independently for the entire five-year duration before moving to the next, whereas in practice a planner would evaluate all parts each day. At the start of any given day, the model's state, which includes on-hand inventory i_d^m and open orders $A_{m,d}^{\text{expected}}$, is updated with realized outcomes. This accounts for both the fulfillment of any demand from the previous day and the addition of any parts that have just arrived.

The core of the model is its methodical and comprehensive exploration of the decision space. This process begins by evaluating the "do nothing" action, which establishes a baseline cost V_0 representing the expected outcome of inaction. This baseline is crucial, as any proactive choice must demonstrate a clear cost improvement over simply letting events unfold. This bounded, full-exploration approach is deliberately more robust than a greedy search; it avoids local optima where an initially promising small action might be inferior to a larger, more beneficial one that a simple incremental step would have missed.

Next, the algorithm explores selling by evaluating every possible sell quantity, from one unit up to the entire on-hand stock. This exploration is governed by two crucial guards that enforce operational and economic prudence. The first, a minimum price threshold, focuses computational effort by ensuring the model only considers economically meaningful sales of higher-value items. The second, a generic risk function, SELLGUARDGENERIC, acts as a strategic constraint. It serves as a safeguard against shortsighted financial optimization, preventing the liquidation of stock that, according to configurable risk metrics, might be critical for maintaining future service levels. The algorithm records the sell quantity that yields the lowest cost, V_{sell}^*.

A similar exhaustive exploration is performed for purchasing. The algorithm evaluates a range of regular order quantities up to a predefined maximum, $Q_{\text{buy}}^{\max}$, which reflects practical constraints such as budget limits or supplier capacity. This search identifies the optimal order size for the day's conditions by finding the global minimum on the cost curve within this defined range. The best cost found is stored as V_{buy}^*.

The final decision is a deterministic comparison of the three mutually exclusive outcomes: waiting (V_0), selling (V_{sell}^*), and buying (V_{buy}^*). The action corresponding to the absolute minimum cost is chosen and its consequences are fully committed to the system's state. It is important to reiterate that express orders are not optimized directly at this stage. They are a secondary consequence determined within the 'ExpectedValue' function, automatically triggered by the projected future state of the winning strategic action. This design ensures all decisions are justified by the same underlying cost model, leading to a consistent and fully traceable procurement strategy. The model's daily choice is therefore deterministic, always selecting the single action with the minimum expected cost. Given a fixed set of demand and lead time realizations, the model would always produce the exact same sequence of decisions.

This daily cost comparison highlights a fundamental aspect of the model's design. Because it makes decisions one day at a time, it is not structured to find a single, globally optimal solution over the entire planning horizon. Its logic is simpler: it just chooses the action with the lowest immediate expected cost. To encourage better long-term financial performance, a manual tuning lever was introduced. The model is calibrated to perceive the cost of a stockout as being significantly higher than its actual financial value. This makes it more cautious and stimulates a drive for better overall results. Through experimentation, setting this perceived penalty to 175% of its true value was found to be the most effective. A lower value resulted in too many stockouts, while a much higher value led to over-procurement and excessive purchase and holding costs. The 175% mark simply represents the calibrated sweet spot for this specific model and dataset.

```
Algorithm 4. Master Procurement model
Input: Sets $M$ and $D$, initial tracking variables $i_0^m$, $A_{m,0}^{\text{expected}}$, demand pmfs $\{\mathcal{E}_l^m\}_{l\in D}$, and all constants
Output: $\{(x_{m,d}^{\text{reg}},\ x_{m,d}^{\text{sell}},\ x_{m,d}^{\text{expr}})\}_{m\in M, d\in D}$
foreach $m \in M$ do
    for $d \in D$ do
        update $i_d^m$, $A_{m,d}^{\text{expected}}$ using previous day records
        // Evaluate case of waiting
        $(V_0,\ A_{m,d}^{(0)},\ I_d^{(0)},\ x_0^{\text{expr}}) \leftarrow \texttt{ExpectedValue}(m, d,\ x_{m,d}^{\text{reg}} = 0,\ x_{m,d}^{\text{sell}} = 0,\ i_d^m,\ A_{m,d}^{\text{expected}})$
        // Full exploration of sell quantities
        $V_{\text{sell}}^{\star} \leftarrow +\infty;\ \ q_{\text{sell}}^{\star} \leftarrow 0;\ \ A_{m,d}^{(\text{sell}\star)} \leftarrow A_{m,d}^{(0)};\ \ I_d^{(\text{sell}\star)} \leftarrow I_d^{(0)};\ \ x_{\text{sell}\star}^{\text{expr}} \leftarrow x_0^{\text{expr}}$
        for $q \in \{0, 1, \ldots, i_d^m\}$ do
            if $P^m \geq SALE\_MIN\_P^m$ and $\textsc{SellGuardGeneric}(m, d, q) = true$ then
                $(V_{\text{sell},q},\ A_{m,d}^{(\text{sell},q)},\ I_d^{(\text{sell},q)},\ x_{\text{sell},q}^{\text{expr}}) \leftarrow \texttt{ExpectedValue}(m, d,\ x_{m,d}^{\text{reg}} = 0,\ x_{m,d}^{\text{sell}} = q,\ i_d^m,\ A_{m,d}^{\text{expected}})$
                if $V_{sell,q} < V_{sell}^{\star}$ then
                    $V_{\text{sell}}^{\star} \leftarrow V_{\text{sell},q};\ \ q_{\text{sell}}^{\star} \leftarrow q;\ \ A_{m,d}^{(\text{sell}\star)} \leftarrow A_{m,d}^{(\text{sell},q)};\ \ I_d^{(\text{sell}\star)} \leftarrow I_d^{(\text{sell},q)};\ \ x_{\text{sell}\star}^{\text{expr}} \leftarrow x_{\text{sell},q}^{\text{expr}}$
        // Full exploration of buy quantities
        $V_{\text{buy}}^{\star} \leftarrow +\infty;\ \ q_{\text{buy}}^{\star} \leftarrow 0;\ \ A_{m,d}^{(\text{buy}\star)} \leftarrow A_{m,d}^{(0)};\ \ I_d^{(\text{buy}\star)} \leftarrow I_d^{(0)};\ \ x_{\text{buy}\star}^{\text{expr}} \leftarrow x_0^{\text{expr}}$
        for $q \in \{1, 2, \ldots, Q_{buy}^{\max}\}$ do
            $(V_{\text{buy},q},\ A_{m,d}^{(\text{buy},q)},\ I_d^{(\text{buy},q)},\ x_{\text{buy},q}^{\text{expr}}) \leftarrow \texttt{ExpectedValue}(m, d,\ x_{m,d}^{\text{reg}} = q,\ x_{m,d}^{\text{sell}} = 0,\ i_d^m,\ A_{m,d}^{\text{expected}})$
            if $V_{buy,q} < V_{buy}^{\star}$ then
                $V_{\text{buy}}^{\star} \leftarrow V_{\text{buy},q};\ \ q_{\text{buy}}^{\star} \leftarrow q;\ \ A_{m,d}^{(\text{buy}\star)} \leftarrow A_{m,d}^{(\text{buy},q)};\ \ I_d^{(\text{buy}\star)} \leftarrow I_d^{(\text{buy},q)};\ \ x_{\text{buy}\star}^{\text{expr}} \leftarrow x_{\text{buy},q}^{\text{expr}}$
        $V^{\text{best}} \leftarrow \min\{\ V_0,\ V_{\text{sell}}^{\star},\ V_{\text{buy}}^{\star}\ \}$
        if $V^{best} = V_0$ then
            $A_{m,d}^{\text{expected}} \leftarrow A_{m,d}^{(0)};\ \ I_d^{m,\text{fore}} \leftarrow I_d^{(0)};\ \ x_{m,d}^{\text{expr}} \leftarrow x_0^{\text{expr}};\ \ x_{m,d}^{\text{sell}} \leftarrow 0;\ \ x_{m,d}^{\text{reg}} \leftarrow 0$
        else if $V^{best} = V_{sell}^{\star}$ then
            $x_{m,d}^{\text{sell}} \leftarrow q_{\text{sell}}^{\star};\ \ i_d^m \leftarrow i_d^m - q_{\text{sell}}^{\star};\ \ A_{m,d}^{\text{expected}} \leftarrow A_{m,d}^{(\text{sell}\star)};\ \ I_d^{m,\text{fore}} \leftarrow I_d^{(\text{sell}\star)};\ \ x_{m,d}^{\text{expr}} \leftarrow x_{\text{sell}\star}^{\text{expr}};$
             $x_{m,d}^{\text{reg}} \leftarrow 0$
        else
            $x_{m,d}^{\text{reg}} \leftarrow q_{\text{buy}}^{\star};\ \ A_{m,d}^{\text{expected}} \leftarrow A_{m,d}^{(\text{buy}\star)};\ \ I_d^{m,\text{fore}} \leftarrow I_d^{(\text{buy}\star)};\ \ x_{m,d}^{\text{expr}} \leftarrow x_{\text{buy}\star}^{\text{expr}};\ \ x_{m,d}^{\text{sell}} \leftarrow 0$
        update $x_{m,d}^{\text{used}}$, $x_{m,d}^{\text{move}}$, $x_{m,d}^{\text{canc}}$
        (re)compute $A_d^{m,\text{fore}}$, $I_d^{m,\text{fore}}$
return $\{x_{m,d}^{reg}, x_{m,d}^{sell}, x_{m,d}^{expr}\}_{m,d}$, $i_d^m$, and refreshed $(A_{m,d}^{expected}, I_d^{m,fore})$ for all $(m, d)$
```

4.3 Demonstration

To make the model's logic more concrete, this section walks through two distinct scenarios for the same part, linked to maintenance task `ZL-533-01-1` (EXP). The first example illustrates a proactive purchase to mitigate a high-probability risk, while the second shows a nuanced decision to accept a small, calculated risk.

For context, the part analyzed has the following characteristics:

- Price: €1,098.17
- Cancellation Penalty: €205,490.85
- Lead Time PMF: day 3 (7.7%), day 4 (23.1%), day 5 (38.5%), day 6 (23.1%), and day 7 (7.7%).

Scenario 1: A Proactive Purchase on Day 281

Context

On day 281, the system has **one unit** of the part in stock. A maintenance event is scheduled for seven days later on day 288. The demand for parts for this task is probabilistic: $\mathbb{P}(\text{Demand} = 0) = 0.651$, $\mathbb{P}(\text{Demand} = 1) = 0.312$, and $\mathbb{P}(\text{Demand} = 2) = 0.037$ (as per Chapter 3).

The Decision

The model evaluates its options as part of its daily routine. The decision here hinges on the 3.7% probability that demand will be two units, which would cause a stockout if the model simply waits with its single unit of inventory. The model must therefore weigh the expected penalty cost from this risk against the certain cost of purchasing an additional part. To do so, the model evaluates the full short-term consequence of each action. This involves exploring every possible scenario by combining all potential demand outcomes for the upcoming task(s) with all possible arrival dates for any existing open orders and potential new order. This evaluation, however, is based only on the action taken today; the model does not simulate or optimize any future procurement decisions. Table 4.1 shows the outcome of this cost comparison.

Table 4.1: Decision Analysis on Day 281

Action	Qty	Value (€)	ST Hold (€)	ST Pur (€)	ST Pen (€)	ST Sell (€)	LT Hold (€)	LT Sell (€)	Notes
buy	1	1,163.27	2.89	1,114.15	0.00	0.00	46.23	0.00	Optimal
buy	2	2,491.73	3.79	2,212.32	0.00	0.00	275.62	0.00	
wait	0	7,734.11	2.00	0.00	7,718.83	0.00	13.28	0.00	Baseline
sell	1	71,309.17	0.00	0.00	72,187.71	878.53	0.00	0.00	Blocked

The values shown for each option is the respective expected cost, which the model calculates by taking a probability-weighted average across all explored future scenarios. The decision centers on the 'wait' and 'buy 1' actions. The 'wait' action's high cost (€7,734.11) is driven almost entirely by the short-term expected cancellation penalty of €7,718.83. In contrast, the 'buy 1' action costs only €1,163.27. By spending €1,114.15 on a purchase, the model completely eliminates the stockout risk. The chosen action is to buy 1.

Scenario 2: Calculated Risk Management on Day 360

Context

On day 360, the situation is more complex. The same task is schedule for **two aircraft** for the next day (day 361), with another single aircraft task following on day 363. This creates a new demand distribution:

$$\mathbb{P}(D=0)=0.424,\quad \mathbb{P}(D=1)=0.406,\quad \mathbb{P}(D=2)=0.146,\quad \mathbb{P}(D=3)=0.023,\quad \mathbb{P}(D=4)=0.001$$

The system starts with **three units** in stock and **one incoming unit** from a previous order. However, this incoming part is not guaranteed to arrive for the task on day 361; it could arrive later, by day 363 (see lead time pmf). Therefore, the model faces a potential shortfall if demand is **four** and the open order is delayed.

The Decision Logic

The model must decide if this timing risk justifies immediate action.

Regular Action: Waiting is Optimal The model's first evaluation (Table 4.2) finds that waiting is the best strategic choice.

Table 4.2: Regular Decision Analysis on Day 360

Action	Qty	Value (€)	ST Hold (€)	ST Pur (€)	ST Pen (€)	ST Sell (€)	LT Hold (€)	LT Sell (€)	Notes
wait	0	360.97	6.05	0.00	368.84	0.00	864.62	-878.53	Baseline
sell*	1	740.42	6.05	1,257.98	368.84	878.53	864.62	-878.53	Blocked
buy	1	947.07	6.95	1,114.15	350.78	0.00	1,232.25	-1,757.07	

*The 'sell' action lowers inventory to a critical level, which auto-activates an express order. The higher cost is due to selling a part only to immediately buy it back at a premium.

The 'wait' action is the clear winner with a total expected cost of €360.97. This value is dominated by the short-term (expected) penalty of €368.84, which correctly prices the risk of the open order arriving late. The model compares this risk to the certain purchase cost of €1,114.15 that would be incurred by the 'buy 1' action. Since the expected penalty is significantly lower than the cost of a new part, the model correctly concludes that it is more economical to accept the calculated risk of a stockout than to make an unnecessary, and far more expensive, purchase.

Express Action: Avoiding Unnecessary Cost Having established waiting as the best strategy, the model confirms that a last-minute express order is not justified (Table 4.3).

Table 4.3: Express Order Decision Analysis on Day 360

Express Qty	Total Value (€)	Purchase (€)	ST Hold (€)	Penalty (€)	LT Hold (€)	LT Sell Rev (€)
0	360.97	0.00	6.05	368.84	864.62	-878.53
1	751.48	1,257.98	8.15	10.78	1,231.63	-1,757.07
2	1,330.08	2,356.15	10.26	0.00	1,599.27	-2,635.60

The cost of ordering one express unit (€751.48) is more than double the cost of waiting (€360.97). The model concludes that the most rational decision is to wait and accept the small timing risk.

4.4 Benchmark

To contextualize and evaluate the performance of the proposed model, its results are compared against two benchmark strategies. These benchmarks are not intended to be sophisticated alternatives but rather to represent two logical solutions in procurement philosophy. The first embodies a highly conservative, risk-averse approach, while the second mimics a common periodic-review policy. By establishing performance bounds with these intuitive strategies, it becomes possible to more clearly assess the specific trade-offs and advantages of the daily, cost-optimization model developed in this thesis. The objective is to understand how the proposed model balances service levels, inventory holding, and operational costs in comparison to simpler, more rigid policies.

4.4.1 Benchmark 1: A Fully Conservative Procurement Strategy

The first benchmark strategy simulates a highly risk-averse policy by making procurement decisions based on worst-case assumptions for both demand and lead times. For any given maintenance task on the horizon, this model assumes that the maximum possible number of units will be required and that any new order placed will take the longest possible time to arrive.

A critical distinction must be made between the simulation environment and the model's decision logic. During the simulation run, the actual realized demand and lead times are still drawn stochastically from their true underlying probability mass functions (PMFs). However, the decision-making agent for this benchmark does not have access to these PMFs. Instead, its view of the future is deterministic: it operates on the belief that the probability of the worst-case scenario (maximum demand, maximum lead time) is 100%. This forces the model to procure enough inventory to cover all conceivable short-term eventualities, aiming for a perfect service level by eliminating stockout risk through proactive over-procurement.

4.4.2 Benchmark 2: A Periodic Review Strategy (12-Month Cycle)

The second benchmark implements a standard periodic review, or ”top-up” policy. This approach reflects a common business practice where inventory is managed through a fixed planning cycle rather than daily adjustments. For this analysis, a review period of 12 months is used.

The mechanics of this strategy are as follows: at the beginning of each 12-month period, a single pre-procurement order is placed for each part. The order quantity is calculated to raise the inventory level to a target: 120% of the total expected demand for that upcoming year. All ordered units are assumed to arrive deterministically at the start of the period. For the remainder of the 12 months, no further regular procurement occurs. The only corrective action available is to place an express order on the day before a scheduled task if a stockout is imminent.

It is important to clarify the role of the 20% safety stock. This buffer is not an additional quantity procured each year. On expectation, the initial safety stock should carry over from one 12-month period to the next. Therefore, the large annual order is primarily meant to replenish the expected consumption for the upcoming year, not to rebuild the safety buffer from scratch. This makes the strategy, in theory, less expensive than the purely conservative approach. This specific target level (20% buffer) is a parameter set for the purposes of this benchmark, rather than a value derived from industry data.

4.4.3 Performance Analysis

The simulation was executed for all three models using a dataset comprising 660 expendable and 414 repairable parts over a five-year horizon. The aggregated results are presented in Table 4.4. The performance metrics are grouped into four categories to facilitate a structured comparison: operational service levels, inventory and supply chain flow, a detailed cost breakdown, and a final financial summary.

Table 4.4: Comprehensive Performance Comparison of Procurement Models

Performance Metric	**Proposed Model**	**Fully Conservative**	**Periodic Review (12-Mo)**
Service & Operational Performance			
Service Level (by days)	99.97%	100.00%	99.98%
Fill Rate (by units)	99.82%	100.00%	99.98%
Fulfilled Task Instances	4,031	4,039	4,035
Cancelled Task Instances	8	0	4
Express Orders (units)	157	0	1,577
Avg. Time in Inventory (days)	251.90	299.57	405.70
Inventory & Supply Flow (in units)			
Opening Inventory	1,074	1,074	1,074
Pre-procurement Arrivals	2,098	2,098	4,025
Repair Returns	1,119	1,119	1,119
Regular Arrivals	3,731	4,705	0
Express Arrivals	157	0	1,577
Total Inflow	**8,179**	**8,996**	**7,795**
Demand Served (Replacements)	4,404	4,407	4,406
Swaps Served (Repairs)	1,170	1,175	1,172
Units Sold	585	0	0
Closing Inventory	2,020	3,414	2,217
Total Outflow	**8,179**	**8,996**	**7,795**
Cost Breakdown (in €)			
Total Purchase Costs	34,994,006.80 (72.9%)	51,108,014.42 (77.6%)	32,673,165.00 (67.8%)
– Pre-procurement	*15,130,714.82*	*15,130,714.82*	*28,163,713.47*
– Regular	*18,352,676.05*	*35,977,299.60*	*0*
– Express	*1,510,615.93*	*0*	*4,509,451.53*
Total Shipping Costs	330,907.66 (0.7%)	270,464.12 (0.4%)	393,453.20 (0.8%)
– Regular	*186,343.30*	*270,464.12*	*0*
– Express	*144,564.36*	*0*	*393,453.20*
Holding Cost	9,174,077.14 (19.1%)	14,507,026.28 (22.0%)	13,631,597.90 (28.3%)
Penalty Cost	3,513,721.50 (7.3%)	0 (0.0%)	1,506,136.76 (3.1%)
Total Operational Costs	**48,012,713.10**	**65,885,504.82**	**48,204,352.86**
Financial Summary (in €)			
Sales Revenue	-7,461,486.29	0	0
Inv. Value Change (Gain (-)/Loss (+))	-6,078,638.51	-24,281,111.96	-10,142,085.32
Final Net Cost*	**34,855,332.60**	**40,968,521.85** (+17.5%)	**38,063,494.90** (+9.2%)

*Final Net Cost is calculated as Total Operational Costs plus/minus Inventory Value Change, less Sales Revenue. Closing inventory is valued at its sell price (80% of purchase cost)

The results presented in the table clearly show that the proposed model achieves the lowest Final Net Cost, outperforming the Fully Conservative strategy by 17.5% and the Periodic Review strategy by 9.2%. While this was the expected outcome, the interesting part is not that it performed better, but how it achieved this result.

A quick look at the service and cost metrics reveals a seemingly counterintuitive finding: the proposed model incurs a significant penalty cost (€3.5 million) resulting from 8 task cancellations, whereas the benchmark models had far fewer or none at all. This is not a sign of failure but is, in fact, evidence of the model's core logic working as intended. In every situation with a potential shortfall, the model weighs the certain cost of purchasing a part against the expected cost of a stockout penalty. The 8 cancellations are simply the few instances where the model took this calculated risk and was unlucky. What is not visible in the table is the far greater number of times the model also chose not to buy a part, and this bet paid off because the bad-case demand did not materialize. It is this ability to consistently avoid unnecessary procurement, at the cost of a few accepted failures, that allows the model to achieve a lower overall net cost.

The inflexibility of the benchmark strategies is also evident in other areas. The Periodic Review model, unable to adapt to demand fluctuations within its 12-month cycle, was forced to rely heavily on expensive express orders, using them to procure 1,577 units. In contrast, the proposed model used express shipping far more strategically, requiring only 157 units as a last resort. The Fully Conservative model, by its nature, avoided all cancellations and express orders, but it paid a steep price in holding costs (€14.5 million), which were over 50% higher than those of the proposed model. This inefficiency is further highlighted by the average time a part spent in inventory, which was nearly 300 days for the conservative model compared to just 252 days for the proposed solution. It is also worth noting that the proposed model was the only one to leverage the selling option, generating over €7.4 million in revenue from 585 units of surplus stock. The impact of this feature will be explored in more detail in the next chapter.

It is important to view these results in the correct context. The absolute monetary values are a direct consequence of the input cost parameters, which are assumptions for this study. The real insight comes from the relative comparison between the models and different input values. The simulation shows how a dynamic, cost-based policy behaves differently from simpler strategies when faced with the same set of uncertain events. The next chapter will build on this by performing a sensitivity analysis on these input parameters. This will help answer important ”what if” questions, such as ”How does the model’s behavior change if penalties are higher?” or ”What is the impact of less reliable suppliers?”. This analysis will further demonstrate the model’s utility as a transparent tool for exploring operational trade-offs.

5 Decision Support Analysis

The previous chapter demonstrated the design and effectiveness of the procurement model, comparing its performance against two simple benchmarks. The primary purpose of this model, however, is not to claim a single, optimal solution. Instead, its true utility lies in its function as a transparent tool for exploration and decision support. Its potential is unlocked by using it to understand why certain outcomes occur and how different variables influence procurement decisions.

The model itself provides benefits on two distinct levels. On an operational level, it gives a planner a tool to understand the immediate trade-offs of a daily decision. For example, they can see exactly how much risk they are taking by choosing to wait on an order versus the certain cost of placing it. This helps procurement move beyond simple intuition and instead base decisions on data. On a tactical and strategic level, managers can use the same model to test future scenarios and shape policy, understanding the financial and service-level consequences of different strategies before committing to them.

The model can now be used as a simulation test bench. By systematically adjusting its core inputs and assumptions, it is possible to isolate and measure the effect of each change on the system's overall performance. This process, a sensitivity analysis, allows for moving beyond a static result to answer critical planning questions about different scenarios.

By observing these effects, the model's outputs can be transformed into actionable insights. The analysis describes the concrete impact of different policies and external factors, making it possible to identify which parameters have the most significant influence on procurement success. This, in turn, shows where management should focus its attention and resources for improvement.

To achieve this, the analysis investigates five key areas. The investigation will explore the impact of supplier reliability through lead time uncertainty and the effect of forecasting accuracy through demand uncertainty. It will also examine the influence of planning horizons, the sensitivity to different cancellation penalties, and the strategic impact of a sales mechanism.

A critical methodological note is required for the following tests. To ensure all comparisons are fair, the simulations are initialized with a consistent random seed. For most scenarios (such as testing penalty costs or sales prices), this means the model faces the exact same sequence of realized demands and supplier arrivals as the baseline run. This consistency ensures that any change in performance is a direct consequence of that parameter change alone. The exceptions, by necessity, are the tests on lead time and demand uncertainty. In those specific cases, altering the underlying probability distributions themselves (e.g., widening the lead time range or biasing the demand forecast) will naturally result in a different sequence of realized events, as the very ”reality” the model faces is what is being tested. This approach is essential for isolating the model's logic. The goal is to understand the patterns of behavior and the trade-offs the model makes, rather than to present statistically averaged absolute numbers.

All simulations use a single and consistent random seed. A single run is not sufficient for statistical robustness, which would require large-scale replications. However, the goal here is not to produce averaged absolute numbers, but rather to isolate the model's logic and clearly observe its behavioral patterns and trade-offs. Averaging the results of many different runs would provide less direct insight to the effect of a specific event on the model's decisions, as the outcomes would be blended. This approach, while less statistically accurate, is therefore more direct with helping the planner's reasoning.

5.1 Lead Time Uncertainty

The first analysis explores one of the most significant external risks in procurement: supplier reliability. This test is designed to isolate the impact of **lead time uncertainty**, not the average lead time. In all scenarios, the expected (mean) and most likely (mode) arrival day for a part remains constant. The mean and mode are 5 and 10 days for expendables and repairables respectively. What changes is the range, or support, of the lead time distribution.

A narrow range (e.g., 3 days for an expendable) implies a highly reliable supplier whose arrivals are very predictable. A wide range (e.g., 9 days for a repairable) represents a highly unreliable supplier, where an order could arrive much earlier or, more critically, much later than expected.

To test this, five scenarios were simulated. The uncertainty range for expendable (EXP) and repairable (REP) parts was progressively widened:

- **Low Uncertainty:** 3-day range (EXP) and 5-day range (REP)
- **Med-Low Uncertainty:** 4-day range (EXP) and 6-day range (REP)
- **Baseline:** 5-day range (EXP) and 7-day range (REP)
- **Med-High Uncertainty:** 6-day range (EXP) and 8-day range (REP)
- **High Uncertainty:** 7-day range (EXP) and 9-day range (REP)

The **Baseline (5/7)** scenario corresponds to the results of the "Proposed Model" presented in the benchmark comparison in the previous chapter. All other parameters, such as costs and demand forecasts, remain identical across all five tests. The aggregated results are presented in Table 5.1.

Table 5.1: Performance Comparison by Lead Time Uncertainty

	Lead Time Range (Expendable / Repairable)				
Performance Metric	**Low (3/5)**	**Med-Low (4/6)**	**Baseline (5/7)**	**Med-High (6/8)**	**High (7/9)**
Service & Operational Performance					
Service Level (by days)	99.97%	99.97%	99.97%	99.97%	99.95%
Fill Rate (by units)	99.82%	99.82%	99.82%	99.82%	99.73%
Cancelled Task Instances	8	8	8	8	12
Express Orders (units)	173	159	157	150	231
Avg. Time in Inventory (days)	251.93	251.41	251.90	252.37	251.33
Inventory Flow (in units)					
Total Procured	5,961	5,958	5,986	5,987	6,078
Units Sold	581	581	585	584	616
Net Inventory Change	+922	+919	+946	+948	+1,005
Cost Breakdown (in €)					
Total Purchase Costs	36,246,772	36,267,729	35,376,751	35,490,506	35,914,068
Total Shipping Costs	379,501	374,449	330,908	324,545	357,333
Holding Cost	9,397,611	9,399,265	9,174,077	9,164,298	9,261,128
Penalty Cost	2,220,140	2,220,140	3,513,722	3,513,722	4,771,171
Total Operational Costs	**48,615,332**	**48,632,890**	**48,871,645**	**48,969,258**	**50,733,143**
Financial Summary (in €)					
Sales Revenue	-7,867,750	-7,869,174	-7,461,486	-7,495,183	-7,149,896
Inventory Value Change	-6,747,373	-6,722,658	-6,554,826	-6,553,172	-6,821,782
Final Net Cost	**34,000,209**	**34,041,059**	**34,855,333**	**34,920,903**	**36,761,465**
vs. Baseline	-2.5%	-2.3%	*Baseline*	+0.2%	+5.5%

The results reveal a clear and direct relationship between supplier uncertainty and total cost. As the lead time range widens, the Final Net Cost steadily increases, rising from €34.00 million in the most predictable scenario (Low 3/5) to €36.76 million in the most uncertain (High 7/9). This confirms that supplier unreliability has a significant and measurable financial impact.

The source of this cost increase is not always obvious. For instance, the number of task cancellations remains static at 8 for all scenarios except the most uncertain one. However, the associated penalty cost is not static. In the two low-uncertainty scenarios, the total penalty is €2.22m, which is significantly lower than the baseline's €3.51m for the exact same number of cancellations. While this data is not sufficient to draw a firm conclusion, it could suggest that with more reliable lead times, the model's calculated risks are more targeted, perhaps becoming more successful at avoiding stockouts on high-value parts and instead sacrificing cheaper ones when necessary.

A more direct driver of cost is the use of express orders. The High Uncertainty (7/9) scenario shows a dramatic spike in express orders (231 units). This increase is often the result of double-ordering for a single requirement. When a regular order fails to arrive in time for its intended task, the model is forced to place a duplicate express order to avoid a stockout. The delayed regular unit eventually arrives and remains in inventory until a future maintenance event requires it. Interestingly, the reverse is not true. The lowest uncertainty (3/5) does not result in the fewest express orders; in fact, it requires more than the baseline. This is an important insight: express orders are not driven only by lead time uncertainty. They are also a critical tool for managing periods of high demand uncertainty, such as when multiple maintenance tasks are scheduled close together.

Across all five tests, the holding costs remain remarkably stable. The inconsistent changes across the different cost categories are a direct result of the model's flexibility. It does not follow fixed rules but constantly re-evaluates the lowest-cost action based on the specific risk profile it faces. The overall conclusion is that while there is a clear positive correlation between lead time uncertainty and total cost, the model adapts to this uncertainty through a dynamic yet transparent mix of actions (accepting penalties, buying express) rather than by just one.

5.2 Demand Uncertainty

Next, this analysis investigates one of the most practical challenges in procurement: making decisions based on imperfect demand forecasts. The model's decisions are only as good as the information it is given. This test is designed to measure the financial consequences of acting on biased information. To do this, the simulation separates the model's perception from reality. The actual realized demand remains the same as in all other tests, but the model is fed forecasts that are intentionally biased. We will explore five scenarios, from a significant underestimation (perceiving 80% of the expected demand) to a significant overestimation (perceiving 120%). The baseline, where the forecast is accurate, is the same "Proposed Model" used in all previous comparisons.

It is important to clarify how this forecast bias is applied. The deviation affects the model's perception at two distinct stages. First, it is applied to the long-term forecast used to determine the initial pre-procurement schedule. This means an overestimation leads to a larger baseline supply, while an underestimation results in a smaller one. Second, the bias is applied daily to the probabilistic demand for individual maintenance tasks. As a task enters the short-term window, the model perceives a shifted probability distribution. For an underestimation, probability mass is shifted towards a demand of zero; for an overestimation, it is shifted away from zero. The results are presented in Table 5.2.

The simulations reveal a fascinating and non-linear relationship between forecast bias and procurement actions. At first glance, the procurement numbers may seem counter-intuitive. For instance, in the -20% underestimation scenario, the model places significantly more regular orders (4,102 units) than in the baseline (3,731 units). Conversely, overestimating demand by +20% leads to the fewest regular orders (3,371 units). Furthermore, the Final Net Cost remained surprisingly stable, with all scenarios

Table 5.2: Performance Comparison by Demand Forecast Accuracy

	Demand Forecast Bias				
Performance Metric	**Under (-20%)**	**Under (-10%)**	**Baseline (0%)**	**Over (+10%)**	**Over (+20%)**
Service & Operational Performance					
Service Level (by days)	99.94%	99.95%	99.97%	99.97%	99.98%
Fill Rate (by units)	99.66%	99.75%	99.82%	99.86%	99.91%
Cancelled Task Instances	15	11	8	6	4
Express Orders (units)	200	182	157	142	142
Avg. Time in Inventory (days)	231.68	244.35	251.90	264.48	281.54
Inventory Flow (in units)					
Total Procured	5,724	5,864	5,986	6,104	6,335
– Pre-procurement	*1,422*	*1,889*	*2,098*	*2,378*	*2,822*
– Regular	*4,102*	*3,793*	*3,731*	*3,584*	*3,371*
– Express	*200*	*182*	*157*	*142*	*142*
Units Sold	444	527	585	642	800
Net Inventory Change	+832	+885	+946	+1,005	+1,076
Cost Breakdown (in €)					
Total Purchase Costs	29,619,875	32,925,345	35,376,751	37,015,053	40,953,379
Total Shipping Costs	384,589	364,340	330,908	341,830	361,672
Holding Cost	8,176,574	8,820,504	9,174,077	9,424,833	10,553,267
Penalty Cost	6,253,861	4,848,422	3,513,722	3,020,258	2,085,769
Inv. Valuation Loss	722,063	556,119	476,187	475,191	475,191
Total Operational Costs	**45,156,961**	**47,514,730**	**48,871,645**	**50,277,165**	**54,429,277**
Financial Summary (in €)					
Sales Revenue	-4,964,534	-6,456,139	-7,461,486	-8,233,961	-11,496,465
Inv. Value Change (Gain (-))	-5,267,516	-6,135,809	-6,554,826	-7,237,308	-7,569,340
Final Net Cost	**34,924,911**	**34,922,781**	**34,855,333**	**34,805,896**	**35,363,472**
vs. Baseline	+0.2%	+0.2%	*Baseline*	-0.1%	+1.5%

landing within 1.5% of the baseline. This is not an error but a clear demonstration of the model's two-level logic, where the daily regular orders are actively compensating for the bias introduced during the initial pre-procurement phase.

This compensation effect is a fascinating insight into the planner's logic, and it is driven almost entirely by the high cost of penalties. The model's primary goal is to reach a "safe" inventory level; a point where it has enough stock to cover an upcoming task and reduce the expected penalty cost to zero. Once it achieves this safe level, the main driver for procurement disappears, and the model "rests". This explains the seemingly paradoxical behavior. In the -20% underestimation case, the initial pre-procurement supply (1,422 units) is far below this safe level. The daily model must then work aggressively, placing a high volume of regular orders (4,102 units) simply to climb up to that minimum safe inventory threshold. Conversely, in the +20% overestimation case, the large pre-procurement (2,822 units) already reaches the safe level, so the daily model has no incentive to buy. It places very few regular orders (3,371 units) and even sells a lot more surplus. The daily orders are simply a corrective tool to get to that penalty-driven "safe" spot.

To make this clearer, the simulation shifted the probabilities of demand outcomes, not the worst-case quantity. This is a crucial distinction. Even in the -20% underestimation case, the planner still knows that a high-demand scenario is possible; it just perceives it as less likely. This is why the penalty-driven logic described above still holds: the model is forced to plan for this worst case, and the bias only makes it slightly more or less forgiving on its risk tolerance. If we had tested a different misspecification, such as underestimating the maximum possible number of items needed for a task, the results would be completely different and likely much worse.

From a financial perspective, acting on biased information is almost always detrimental. Underestimating demand is clearly damaging; the -20% scenario causes penalty costs to nearly double, spiking to €6.25 million and increasing the final net cost. Overestimating demand reveals a more complex trade-off. A slight overestimation (+10%) can be marginally beneficial (a 0.1% cost reduction in this case), as the savings from fewer cancellations just outweigh the cost of buying and carrying extra inventory. This benefit quickly disappears, however. In the +20% scenario, the cost of procuring and holding the large surplus (driving holding costs over €10.5 million) becomes too great, causing the final net cost to rise by 1.5%. This suggests that if the forecast is biased, the failure probability function should be adjusted using operational observations, but only once statistically sound proof is gathered.

5.3 Short-Term Forecast Horizon

This test focuses on a core parameter of the model's design: the length of the short-term forecast window. As described in Algorithm 3, the model operates with a dual horizon. It evaluates the immediate future (the "short-term") by building a full, detailed tree of all possible probabilistic scenarios. Beyond this window, it switches to a less computationally intensive approximation (the "long-term tail") that relies on moment propagation.

By default, the length of this high-fidelity short-term window is set for each part as the maximum possible lead time for a new regular order. This ensures the model always performs a detailed analysis to cover the immediate uncertainty regarding when a new order might arrive. This section tests what happens if we extend this window significantly. It is important to note that this practically means that the model or the planner is called to handle the procurement of parts for a specific task on an earlier date than needed.

At first, extending this high-detail window might seem like it could only improve results, or at worst, do no harm. In reality, this change could have a complex trade-off. A longer analytic window forces the model to "see" and calculate the cost of distant, potential risks with high precision. This might cause it to commit to a purchase decision much earlier than necessary, when there was still time to wait for more information (like demand materializing) before acting. This test is designed to see if that forced early commitment is beneficial or costly.

To test this, the short-term window was set to a fixed **21 days** for all parts. This value is computationally feasible but also much longer than the worst-case lead times for both expendable and repairable parts, giving the model a much longer, high-fidelity view of the near future. All other parameters are identical to the baseline. The results are compared in Table 5.3.

The results show that acting "too soon" is not always better, as forcing a longer look-ahead resulted in a 2.3% increase in the final net cost. This extra cost came from committing to decisions earlier, which directly increased both total purchase costs and holding cost. The impact on holding cost is clearly visible in the average time in inventory, which rose by nearly 14 days. On the positive side, this proactive strategy did make operations less reactive, cutting the number of expensive express orders almost in half (from 157 to 80). However, this came at the cost of a large increase in both purchased and sold units, suggesting a higher volume of logistical activity.

This demonstrates a key insight for planners: acting too early on uncertain, long-range forecasts can lead to costly regret, as new information gathered well before the maintenance task often reveals the initial purchase was unnecessary. This unnecessary buy/sell activity adds costs without providing a significant service benefit, as the total penalty cost was almost identical to the baseline despite one fewer cancellation. Sometimes, demand is simply unforgiving, and the high cost of insuring against every potential, distant stockout is not worth the investment. Waiting for more information is often the more economical choice.

Table 5.3: Performance Comparison by Forecast Horizon Length

Performance Metric	Baseline (Dynamic Lead Time)	Extended Window (21 Days)
Service & Operational Performance		
Service Level (by days)	99.97%	99.97%
Fill Rate (by units)	99.82%	99.84%
Cancelled Task Instances	8	7
Express Orders (units)	157	80
Avg. Time in Inventory (days)	251.90	265.27
Inventory Flow (in units)		
Total Procured (Pre, Reg, Exp)	5,986	6,278
Units Sold	585	708
Net Inventory Change	+946	+1,115
Cost Breakdown (in €)		
Total Purchase Costs	35,376,751	37,021,092
Total Shipping Costs	330,908	293,339
Holding Cost	9,174,077	9,635,064
Penalty Cost	3,513,722	3,470,121
Total Operational Costs	**48,871,645**	**50,419,616**
Financial Summary (in €)		
Sales Revenue	-7,461,486	-8,073,230
Inv. Value Change (Gain (-)/Loss (+))	-6,554,826	-6,678,476
Final Net Cost	**34,855,333**	**35,667,909**
vs. Baseline	*Baseline*	+2.3%

5.4 Penalty Costs

The cost-driven strategy, as seen in the benchmark comparison, is not designed to be the most conservative. It works by accepting some calculated risks, and this entire trade-off balances on a single, crucial number: the monetary penalty assigned to a stockout. This value is the planner's financial estimate for all the operational chaos that follows a cancellation, from schedule delays to a grounded aircraft.

Since this penalty is an estimate of a complex reality, it can be misjudged. This analysis explores the practical consequences of under- or overestimating this value. It provides insight into how procurement behavior, service levels, and total costs change as this penalty estimate shifts. This helps to understand just how sensitive the procurement plan is to this one financial assumption. The results of this analysis are summarized in Table 5.4

The baseline configuration for this study used a penalty of €200,000 plus five times the part's price (P). To test the sensitivity to this estimate, four new scenarios were simulated. These scenarios represent a wide range of risk profiles, from a low-penalty estimate (€50k + 2P) that is more tolerant of stockouts, up to a very high-penalty estimate (€1M + 7P) that treats any cancellation as a severe financial event. All other model parameters remain identical to the baseline.

A lower penalty estimate leads to the lowest Final Net Cost, but this comes at the cost of service. The "Low" and "Med-Low" scenarios, which are 9.5% and 5.2% cheaper than the baseline, also see a sharp increase in task cancellations (to 23 and 14, respectively). From a purely financial perspective, this strategy is logical: the savings from reduced procurement and holding costs are greater than the cost of the penalties incurred. More interesting is the trend at the high end. As the penalty estimate increases from the baseline to "Med-High," the net cost also rises by 2.6%. Pushing the estimate even further to "High," however, causes the net cost to drop back slightly (to +0.9%). This illustrates how a very high penalty estimate forces a more proactive procurement strategy. This proactive stance is expensive,

Table 5.4: Performance Comparison by Cancellation Penalty Estimate

	Penalty Structure (Fixed + Variable)				
Performance Metric	**Low (50k + 2P)**	**Med-Low (100k + 3P)**	**Baseline (200k + 5P)**	**Med-High (400k + 6P)**	**High (1M + 7P)**
Service & Operational Performance					
Service Level (by days)	99.90%	99.94%	99.97%	99.98%	99.99%
Fill Rate (by units)	99.46%	99.68%	99.82%	99.89%	99.98%
Cancelled Task Instances	23	14	8	5	1
Express Orders (units)	180	156	157	158	161
Avg. Time in Inventory (days)	246.36	246.20	251.90	260.66	268.96
Inventory Flow (in units)					
Total Procured	5,684	5,811	5,986	6,097	6,219
Units Sold	476	504	585	652	719
Net Inventory Change	+769	+858	+946	+987	+1,038
Cost Breakdown (in €)					
Total Purchase Costs	28,255,813	30,650,094	35,376,751	36,749,583	38,606,641
Total Shipping Costs	306,931	323,867	330,908	327,312	338,390
Holding Cost	7,973,496	8,460,479	9,174,077	9,584,363	10,311,442
Penalty Cost	3,813,350	4,103,385	3,513,722	3,757,982	1,951,805
Total Operational Costs	**40,349,591**	**43,537,825**	**48,871,645**	**50,419,240**	**51,208,278**
Financial Summary (in €)					
Sales Revenue	-5,452,105	-5,999,636	-7,461,486	-8,204,272	-9,214,952
Inv. Value Change (Gain (-))	-3,350,932	-4,512,016	-6,554,826	-6,461,257	-6,813,554
Final Net Cost	**31,546,554**	**33,026,174**	**34,855,333**	**35,753,710**	**35,179,773**
vs. Baseline	-9.5%	-5.2%	*Baseline*	+2.6%	+0.9%

driving up both purchase and holding costs, but it is so cautious that it almost entirely eliminates cancellations, dropping them to just one. This shows a clear and non-linear balance between paying for inventory upfront or paying for service cancellations later.

So far we analysed the actual penalty, affecting both the decision logic and the final cost. A second test explores a different lever: the perceived penalty. As noted, the baseline strategy is calibrated to be cautious by perceiving any stockout penalty as 175% of its true value (a 1.75× multiplier). This test explores the effect of that "caution" buffer. We will compare the 1.75× baseline against a 1.0× scenario (where decisions are based on the true penalty), and two scenarios with increased caution (2.0× and 2.5×). In all cases, the actual penalty cost charged for a cancellation remains the standard baseline value ($200\text{k} + 5P$). Results are depicted in Table 5.5.

The level of caution built into the procurement strategy is a powerful lever. Acting with no additional caution (the 1.0× scenario) is not the best approach; it results in a 2.8% higher net cost compared to the baseline, driven by a high number of cancellations (14) and the resulting €6.18 million in penalties. Intentionally overestimating the penalty is clearly beneficial, but only to a point. A 2.0× caution multiplier achieves the lowest final net cost (2.2% below baseline) by cutting cancellations to just 6. This benefit has a limit, however. Pushing the caution even further to 2.5× increases total purchase and holding costs so much that the extra spending is not fully offset by the small gain of one fewer cancellation, causing the net cost to rise again. This suggests a "sweet spot" for caution, where the current cost structure finds a natural balance between proactive procurement and service failures.

Table 5.5: Performance Comparison by Perceived Penalty Multiplier

	Perceived Penalty Multiplier			
Performance Metric	**1.0x (No Overestimation)**	**1.75x (Baseline)**	**2.0x (High Caution)**	**2.5x (V. High Caution)**
Service & Operational Performance				
Service Level (by days)	99.94%	99.97%	99.97%	99.98%
Fill Rate (by units)	99.68%	99.82%	99.86%	99.89%
Cancelled Task Instances	14	8	6	5
Express Orders (units)	158	157	150	151
Avg. Time in Inventory (days)	246.46	251.90	253.06	256.99
Inventory Flow (in units)				
Total Procured	5,818	5,986	6,007	6,049
Units Sold	501	585	591	619
Net Inventory Change	+868	+946	+959	+972
Cost Breakdown (in €)				
Total Purchase Costs	31,412,091	35,376,751	35,610,804	36,656,313
Total Shipping Costs	319,447	330,908	322,392	316,813
Holding Cost	8,468,444	9,174,077	9,285,243	9,459,632
Penalty Cost	6,179,231	3,513,722	2,485,949	2,255,701
Total Operational Costs	**46,379,213**	**48,871,645**	**47,704,388**	**48,688,459**
Financial Summary (in €)				
Sales Revenue	-6,014,224	-7,461,486	-7,406,134	-7,906,491
Inv. Value Change (Gain (-))	-4,530,268	-6,554,826	-6,215,867	-6,412,763
Final Net Cost	**35,834,721**	**34,855,333**	**34,082,387**	**34,369,205**
vs. Baseline	+2.8%	*Baseline*	-2.2%	-1.3%

5.5 Sales

A unique feature of this research is the exploration of the ability to sell inventory when judged appropriate. The analysis in this section explores how that capability influences procurement behavior. The impact is not just the revenue from the sales themselves. More importantly, the option to sell in the future changes the cost-benefit calculation for every purchase decision made today. A high recovery price makes proactive purchasing less costly. The model can be encouraged to buy a part to cover a potential shortfall, knowing that if the part is not needed, a large portion of its cost can be recovered. This is a key difference from a model that simply accounts for final inventory value, as the selling option is an active part of the daily cost-minimization logic.

To measure this effect, we will compare the baseline (which uses an 80% recovery price) against three new scenarios: one where the selling mechanism is disabled entirely (”No Sales”), one with a lower recovery value (70%), and one with a higher value (90%). A value near 100% is not tested because it would break the logic, creating a ”free” return policy that would encourage the model to over-procure and just return the excess. These different price points represent the operational ”friction” of selling; a 70% price implies a higher cost or effort to sell a part, while a 90% price makes it an easy, low-cost option. The results are shown in Table 5.6. Note that the selling mechanism is active only for items with purchase price above €500.

Table 5.6 illustrates how the option to sell affects the procurement strategy and the financial outcome. Comparing the scenarios to the standard ”No Sales” approach, it is clear the benefit is not automatic. For instance, both the 70% and 80% (Baseline) recovery price scenarios result in a higher net cost than not integrating sales at all. This suggests that a poorly managed or inefficient selling mechanism can

Table 5.6: Performance Comparison by Selling Price

	Selling Price (as % of Purchase Price)			
Performance Metric	**No Sales**	**Sell at 70%**	**Baseline (Sell at 80%)**	**Sell at 90%**
Service & Operational Performance				
Service Level (by days)	99.96%	99.96%	99.97%	99.97%
Fill Rate (by units)	99.77%	99.80%	99.82%	99.84%
Cancelled Task Instances	10	9	8	7
Express Orders (units)	133	151	157	200
Avg. Time in Inventory (days)	247.18	250.18	251.90	257.01
Inventory Flow (in units)				
Total Procured	5,657	5,871	5,986	6,191
Units Sold	0	495	585	765
Net Inventory Change	+1,204	+922	+946	+970
Cost Breakdown (in €)				
Total Purchase Costs	28,254,680	32,066,323	35,376,751	38,173,001
Total Shipping Costs	263,340	315,704	330,908	370,015
Holding Cost	8,668,755	8,910,284	9,174,077	9,565,047
Penalty Cost	4,694,461	4,358,572	3,513,722	2,799,137
Total Operational Costs	**43,559,913**	**46,781,520**	**48,871,645**	**51,722,019**
Financial Summary (in €)				
Sales Revenue	0	-5,416,001	-7,461,486	-11,216,874
Inv. Value Change (Gain (-))	-9,264,844	-5,181,134	-6,554,826	-7,812,772
Final Net Cost	**34,295,068**	**36,184,384**	**34,855,333**	**32,692,372**
vs. Baseline	-1.6%	+3.8%	*Baseline*	-6.2%

be worse than having no mechanism, and that its financial benefit is only unlocked at a very high level of efficiency.

This behavior is driven by the model's cost calculation. The potential to sell a part in the future acts as a discount on its initial purchase, lowering the perceived cost of procurement. As the selling price increases, the model becomes more proactive, visible in the rising total procured units (from 5,657 in the "No Sales" case to 6,191 at 90%). This more aggressive procurement strategy directly improves service levels, cutting cancelled task instances from 10 down to 7 and reducing the associated penalty cost from €4.7m to €2.8m. This clearly illustrates the balance between procurement costs and the resulting service failures.

However, this strategy is a trade-off that is highly sensitive to price. The "sell at 90%" scenario is the only one where this trade-off is clearly profitable, achieving a 6.2% net cost reduction compared to the baseline. Here, the high sales revenue (€11.2 million) and lower penalties are more than enough to offset the increased total purchase costs and holding cost. In contrast, the "sell at 70%" scenario shows the risk of this strategy: the model still procures more (and thus decreases chance of stockout), but the low recovery price is not enough to cover that investment, resulting in the highest net cost of all four cases (+3.8% vs. baseline). Therefore, enabling a sales mechanism is only advisable if the operational costs of selling are low enough to guarantee a high recovery price.

6 Conclusion

This thesis presents a framework designed to bridge the persistent gap between a fixed, fleet-wide maintenance schedule and the complex, day-to-day procurement decisions required to support it. Traditional, aggregate-level planning often fails to capture the specific needs of individual checks, leading to inefficient inventory. This work moved beyond that by focusing on a granular approach, where material requirements are derived directly from each individual, scheduled maintenance task, linking demand directly to the operational calendar.

The resulting procurement model is designed for operational utility, where traceability is a central feature. All decisions, including both purchasing new parts and selling surplus stock, are governed by a transparent, cost-based logic. This emphasis on explainability is a deliberate choice. In an environment with incomplete data, the primary goal is to provide planners with a tool that generates actionable insights and reveals the ”why” behind a recommendation, which is often more valuable than a ”black box” claim of optimality.

To tackle this problem, the research was structured into three distinct, sequential stages. The first stage involved developing a demand forecast. This required building a methodology to convert the raw maintenance plan into a time-phased, probabilistic demand signal for individual spare parts. The second stage was to create a procurement model that could act on this detailed forecast, making daily purchasing and inventory decisions to minimize total cost under uncertainty. The final stage focused on validation and analysis, first by testing the model against logical benchmarks and then by using it as a tool to generate practical insights for planners.

This three-stage process functions as a modular pipeline, where each component is self-contained and can be adjusted or even swapped out to suit different operational needs. The framework is not rigid; its components are interchangeable. For example, the task clustering method used in forecasting could be replaced with a different technique, such as a decision tree, without disrupting the overall flow. Similarly, the procurement model itself could be adapted, perhaps by aggregating decisions to a component-group level instead of a part-specific one, or by expanding it to optimize procurement across all parts simultaneously. This flexibility is key to its practical application.

The forecasting component first had to solve a fundamental data problem: the maintenance plan's raw task descriptions are not directly usable for demand planning. To overcome this, the Maintenance Planning Document (MPD) was used as a foundation to systematically group tasks based on their technical attributes. This grouping allowed for a structured, model-based assignment of failure probabilities to each task cluster. It is important to state that neither the clusters themselves nor the synthetic reliability models were validated against real-world data. However, the framework successfully demonstrated that it is possible to move beyond aggregate forecasts. It provides a systematic and repeatable method for describing the demand potential of each individual maintenance task, laying a foundation that can be readily improved with expert input or historical failure data.

The procurement algorithm was built to replicate the thought process of a human planner, where every decision is a transparent, trade-off-based balance. Instead of an opaque optimization, the model operates on a daily basis by comparing the expected future costs of buying, waiting, or selling. To mirror a realistic operational environment, the framework includes three distinct procurement mechanisms: a baseline pre-procurement schedule for long-term needs, flexible regular orders with uncertain lead times, and reactive express orders for imminent shortfalls. This design ensures that every action is a justifiable choice between competing costs, such as holding inventory versus risking a stockout.

This model was tested against two simple benchmarks: a fully conservative strategy aiming for 100% service and a common periodic review policy. The proposed model performed 17.5% and 9.2% better, respectively, in terms of total net cost. This numerical victory is not the central achievement, as these benchmarks are by nature naive and inflexible. The more valuable insight comes from how the model performed better. The most significant conclusion is that allowing for a small, controlled number of risks (cancellations) is not only justified but financially optimal when the cost of perfect protection is too high. Secondly, the results confirm the clear advantage of treating each task independently and using all available information, rather than relying on static, high-level rules.

This task-specific approach also addresses a wider challenge in aviation management. Just as maintenance planning has moved toward date-specific operations, procurement must follow suit to support the tighter schedules of Equalized Maintenance Programs. For an EMP to function, every supporting element must operate with the same level of detail. Without a granular link between tasks and materials, the supply chain becomes a weak point that can compromise the stability of the entire equalized schedule.

Further analysis of the procurement drivers, conducted by isolating key variables, reveals several practical insights for planners. While some relationships are intuitive, such as greater supplier lead time uncertainty correlating with higher net costs, the model's main utility is its ability to navigate complex, non-linear trade-offs. The results demonstrate that not all operational variables affect the system in a predictable way, reinforcing the need for a decision support tool that can manage these nuanced interactions rather than relying on static rules.

Certain findings from the analysis are direct: the ability to sell surplus stock, for example, is only financially beneficial when it can be executed efficiently with a high recovery price. Otherwise, the option is detrimental. Other results, however, point to a more nuanced 'gray area' in the decision space. Within this region, changes to key parameters, such as the perceived 'caution' for penalties, do not linearly improve or degrade performance. Instead, this region is characterized by an operational balance where an increase in procurement investment is offset by a nearly equal decrease in penalty costs, holding the net outcome stable. This suggests that performance degrades significantly only when decisions fall far outside of this balanced region.

Ultimately, these trade-offs all center on the fundamental procurement problem: minimizing total costs while maintaining a sufficient service level. By translating the operational impact of a service failure into a monetary penalty, the strategy becomes a transparent financial trade-off. A key insight from this work is observing how robustly the cost-driven model manages this balance. This trade-off is between investing in procurement and holding costs upfront, versus risking a significantly higher penalty cost later. The model's logic proved robust even under poor forecasts; when demand was underestimated, the daily function still compensated by increasing orders to maintain a 'safe' inventory level, driven by the high financial weight of a potential stockout.

7 Discussion & Future Work

The framework in this thesis was built to tackle a complex, real-world problem. To deliver a complete end-to-end solution within the research scope, several boundaries and assumptions were necessary. These choices helped to focus the preceding analysis and also provide a natural starting point for looking ahead. Reflecting on these limitations is valuable, as it establishes a practical foundation for future extensions and new lines of inquiry.

Validating every parameter in a model like this is a major challenge. In aviation, it's even tougher because detailed data on component failures or supplier performance is often proprietary and hard to get from the outside. This meant that assumptions were needed to build parts of the model, especially for the demand forecast. The framework was built to handle this reality by focusing on transparency. The procurement model, for instance, is built on explicit cost levers and traceable decisions. This design makes sure that even when an input is an assumption, its specific impact on a final recommendation is still clear and can be checked.

A key area for future improvement is the demand forecast element, especially the task clustering. It was hard to tell which clustering method was truly better without real-world data for validation. This challenge, however, points to an important solution, one that does not depend solely on big datasets. Expert opinion, especially from ground engineers, could be just as valuable, or even more so. They have the hands-on experience and are arguably the best people to group tasks based on the simple question of "how often does this job lead to a replacement?"

The procurement model handles every part individually, which offers a much deeper level of control than traditional, high-level planning. This part-specific logic could be developed even further, perhaps using a decision-tree structure to apply different management rules based on a part's cost, criticality, or supply risk. At the same time, this individual focus misses a clear opportunity: economies of scale. A valuable next step would be to look at procurement system-wide, allowing the model to group orders for different parts that come from the same supplier. This higher-level view would also be a natural place to integrate key performance indicators (KPIs), letting a planner make informed trade-offs, such as accepting a small risk on one task to guarantee the success of a more critical one.

Future work could also address parameter uncertainty by adopting a robust optimization mindset. While this framework relies on specific estimates for inputs like penalty costs or lead time agreements, an alternative approach could solve for a range of potential values. By optimizing for the worst-case scenario, this method would ensure that the trade-off between procurement and risk remains valid regardless of exact parameters, offering reliable decision support even when specific operational data is effectively unavailable.

Furthermore, while this framework focuses on planned maintenance, a natural extension is to integrate unplanned, corrective maintenance. The current model can already handle these events reactively: a planner could create a small buffer by manually increasing failure probabilities, and the model would adapt to the sudden drop in stock when an unplanned part is used. A more systematic solution, however, would be to build this uncertainty directly into the forecast. This could be done by adding a low-level, daily failure probability for critical parts, which would then be included in the model's short-term cost evaluation, allowing it to balance both planned and unplanned risks.

The sensitivity analysis in this report was used to understand the model's behavior by testing one parameter at a time. This same framework could be used to pursue a different goal: finding the absolute best operational or financial results. To do this, one could run mass simulations, testing thousands of different scenarios and combinations of parameters. This would allow for a complete fine-tuning of all the economic levers, such as penalty costs and caution multipliers, to find the most effective and profitable policy for a specific set of operational goals.

Bibliography

Ayu Nariswari, N. P., Bamford, D., and Dehe, B. (2019). Testing an ahp model for aircraft spare parts. *Production Planning & Control*, 30(4):329–344.

Bacchetti, A. and Saccani, N. (2012). Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. *Omega*, 40(6):722–737.

Bokinsky, H., McKenzie, A., Bayoumi, A., McCaslin, R., Patterson, A., Matthews, M., Schmidley, J., and Eisner, L. (2013). Application of natural language processing techniques to marine v-22 maintenance data for populating a cbm-oriented database. In *AHS Airworthiness, CBM, and HUMS Specialists' Meeting, Huntsville, AL*.

Buyukkaramikli, N. C., van Ooijen, H. P., and Bertrand, J. W. M. (2015). Integrating inventory control and capacity management at a maintenance service provider. *Annals of Operations Research*, 231:185–206.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303.

Deshpande, V., Iyer, A. V., and Cho, R. (2006). Efficient supply chain management at the us coast guard using part-age dependent supply replenishment policies. *Operations Research*, 54(6):1028–1040.

Dinh, T., Wong, H., Fournier-Viger, P., Lisik, D., Ha, M.-Q., Dam, H.-C., and Huynh, V.-N. (2025). Categorical data clustering: 25 years beyond k-modes. *Expert Systems with Applications*, 272:126608.

Erkoc, M. and Ertogral, K. (2016). Overhaul planning and exchange scheduling for maintenance services with rotable inventory and limited processing capacity. *Computers & Industrial Engineering*, 98:30–39.

Guha, S., Rastogi, R., and Shim, K. (2000). Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366.

Hekimoğlu, M., Kök, A. G., and Şahin, M. (2022). Stockout risk estimation and expediting for repairable spare parts. *Computers & Operations Research*, 138:105562.

Hu, Q., Bai, Y., Zhao, J., and Cao, W. (2015). Modeling spare parts demands forecast under two-dimensional preventive maintenance policy. *Mathematical Problems in Engineering*, 2015(1):728241.

International Air Transport Association (2015). *Guidance Material and Best Practices for Inventory Management*. International Air Transport Association, Montreal, Geneva, 2nd edition.

Kang, Z., Marandi, A., Basten, R. J., and de Kok, T. (2023). Robust spare parts inventory management. *Available at SSRN 4553430*.

Kontrec, N., Petrović, M., Vujaković, J., and Milošević, H. (2016). Implementation of weibull's model for determination of aircraft's parts reliability and spare parts forecast. In *Mathematical and Information Technologies, MIT-2016, CEUR Workshop Proceedings*.

Li, P., Wen, M., Zu, T., and Kang, R. (2023). A joint location–allocation–inventory spare part optimization model for base-level support system with uncertain demands. *Axioms*, 12(1):46.

Liu, Y., Feng, Y., Xue, X., and Lu, C. (2018). Research on multi-echelon inventory system for civil aircraft spare parts with lateral transshipments and importance degree. In *2018 12th international conference on reliability, maintainability, and safety (ICRMS)*, pages 434–441. IEEE.

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

Nanyonga, A., Joiner, K., Turhan, U., and Wild, G. (2025). Applications of natural language processing in aviation safety: A review and qualitative analysis. In *AIAA SciTech 2025 Forum*, page 2153.

Nguyen, H. H. (2017). Clustering categorical data using community detection techniques. *Computational intelligence and neuroscience*, 2017(1):8986360.

Obradovic, G. (2023). *Mixed-integer optimization modeling for the simultaneous scheduling of component replacement and repair*. Chalmers Tekniska Hogskola (Sweden).

Pham, H., editor (2006). *Springer Handbook of Engineering Statistics*. Springer, London.

Qin, Y., Ma, H.-L., Chan, F. T., and Khan, W. A. (2020). A scenario-based stochastic programming approach for aircraft expendable and rotable spare parts planning in mro provider. *Industrial Management & Data Systems*, 120(9):1635–1657.

Roda, I., Macchi, M., Fumagalli, L., and Viveros, P. (2014). A review of multi-criteria classification of spare parts: From literature analysis to industrial evidences. *Journal of Manufacturing Technology Management*, 25(4):528–549.

Romeijnders, W., Teunter, R., and Van Jaarsveld, W. (2012). A two-step method for forecasting spare parts demand using information on component repairs. *European Journal of Operational Research*, 220(2):386–393.

Sinha, P., Calfee, C. S., and Delucchi, K. L. (2021). Practitioner's guide to latent class analysis: methodological considerations and common pitfalls. *Critical care medicine*, 49(1):e63–e79.

Stranieri, F., Fadda, E., and Stella, F. (2024). Combining deep reinforcement learning and multi-stage stochastic programming to address the supply chain inventory management problem. *International Journal of Production Economics*, 268:109099.

Syntetos, A. A. and Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of forecasting*, 21(2):303–314.

te Booij, J. (2024). *Prediction of aircraft component failure using smart feature engineered models*. PhD thesis, Tilburg University.

Tiemessen, H. G., Fleischmann, M., and Van Houtum, G.-J. (2017). Dynamic control in multi-item production/inventory systems. *OR spectrum*, 39(1):165–191.

Usuga-Cadavid, J. P., Lamouri, S., Grabot, B., and Fortin, A. (2022). Using deep learning to value free-form text data for predictive maintenance. *International Journal of Production Research*, 60(14):4548–4575.

Van der Auweraer, S. and Boute, R. (2019). Forecasting spare part demand using service maintenance information. *International Journal of Production Economics*, 213:138–149.

Verhagen, W. J. and De Boer, L. W. (2018). Predictive maintenance for aircraft components using proportional hazard models. *Journal of Industrial Information Integration*, 12:23–30.

Wang, W. and Syntetos, A. A. (2011). Spare parts demand: Linking forecasting to equipment maintenance. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1194–1209.

Zhang, S., Huang, K., and Yuan, Y. (2021). Spare parts inventory management: A literature review. *Sustainability*, 13(5):2460.

Zhu, S., van Jaarsveld, W., and Dekker, R. (2020). Spare parts inventory control based on maintenance planning. *Reliability Engineering & System Safety*, 193:106600.

A Maintenance Planning Document (MPD)

The Maintenance Planning Document (MPD) is a foundational document published by the aircraft manufacturer and serves as an envelope repository for repetitive maintenance tasks. Its primary purpose is to provide a basis for aircraft operators to develop their own customized and locally approved scheduled maintenance programs. The MPD is not a standalone regulation; rather, it consolidates requirements from several official source documents, including the Maintenance Review Board Report (MRBR), the Airworthiness Limitation Section (ALS) Parts 2, 3, 4, and 5, and the ETOPS Configuration, Maintenance and Procedures (CMP) Document.

Tasks originating from the MRBR and ALS are declared as Instructions for Continued Airworthiness (ICA). These tasks constitute the initial minimum set of requirements that must be included in an operator's maintenance program when an aircraft enters service. This thesis uses the Maintenance Planning Document (MPD) for the A320 family, revision 26. While the MPD is a comprehensive guide, operators remain responsible for independently ensuring their maintenance programs comply with all mandatory requirements, such as Airworthiness Directives (ADs) and the entirety of the ALS.

MPD Data Structure

The MPD presents maintenance tasks in a structured tabular format. Each row corresponds to a unique maintenance task, and each column provides specific data. The following descriptions detail all the columns, with expanded detail for those most relevant to spare parts forecasting.

Revision Code. A simple marker (N=New, R=Revised, D=Deleted) indicating changes from the previous MPD revision.

Task Number. Each task is identified by a unique 9 to 10-digit number (e.g., `XXXXXX-XX-X`). This is a composite identifier where the first six digits typically refer to the ATA chapter or aircraft zone, the next two digits are a sequence number, and the final digit(s) serve as an applicability index for different aircraft configurations.

Zone. This column specifies the physical location on the aircraft where the task is to be performed. Zones are identified by standardized three-digit numbers that correspond to specific areas of the airframe (e.g., fuselage sections, wings, landing gear bays). A task may list multiple zones, indicating that the action must be repeated in each location.

Description. This is a critical column that provides a concise, free-text summary of the work to be performed. It identifies the system, component, or structural item and the nature of the task. This field often contains the most detailed technical information available within the MPD and is essential for understanding the potential need for spare parts. The column also includes standardized codes for the required skill (e.g., AF for Airframe, AV for Avionics).

Task Code. This provides a standardized abbreviation for the type of work required, allowing for quick identification and grouping of similar maintenance actions. Key task codes include:

- **CHK**: Check. A quantitative check to verify if an item is within specified limits.
- **GVI**: General Visual Inspection. A visual survey to detect obvious damage, failure, or irregularity.
- **CLN**: Clean. The removal of contaminants from a component or area.
- **DS**: Discard. Mandatory removal and replacement of an item at its life limit.

- **OP**: Operational Check. A qualitative check to confirm a system is functioning correctly.
- **DI**: Detailed Inspection. An intensive, close-up visual examination to detect damage or failure.
- **RS**: Restoration. Work required to return an item to a defined standard (e.g., overhaul).
- **VC**: Visual Check. A visual check to confirm an item is present and correctly positioned.
- **FC**: Functional Check. A quantitative check to verify a system performs within measured limits.
- **TPS**: Test/Performance Check. A test against a specific technical or performance standard.
- **RAR**: Repair/Assessment/Reporting. Repair, assessment, or reporting actions based on findings.
- **LU**: Lubrication. The application of lubricants to reduce friction between moving parts.
- **SV**: Servicing. Replenishing fluids or gases to keep equipment operational.
- **SDI**: Special Detailed Inspection. An intensive inspection requiring specialized techniques or equipment.
- **BSI**: Borescope Inspection. An internal inspection using a borescope to avoid disassembly.
- **OPT**: Optional. A non-mandatory task performed at the operator's discretion.

TCI / RSC. Time Controlled Items (TCI) and Removable Structural Components (RSC) are markers used as planning aids to identify components that are typically removed and managed as rotable assets.

Interval and Threshold. This column defines the scheduling requirements for the task. It consists of two main components: a **Threshold (T)** for the first accomplishment and an **Interval (I)** for subsequent accomplishments. These limits are expressed in usage parameters, and the one that occurs first dictates the schedule. The units used are Flight Hours (FH), Flight Cycles (FC), Days (DY), Months (MO), and Years (YE).

Source. The Source column indicates the origin document of the maintenance requirement, which is crucial for understanding its regulatory weight and criticality.

- **MRB**: A task from the Maintenance Review Board, categorized by a Failure Effect Category (FEC) number. This number indicates the consequence of a potential failure (5: Evident Safety, 6: Evident Operational, 7: Evident Economic, 8: Hidden Safety, 9: Hidden Economic).
- **ALI**: Airworthiness Limitation Item from ALS Part 2. Mandatory inspections for fatigue-critical structures.
- **CMR* / CMR****: Certification Maintenance Requirement from ALS Part 3. Mandatory tasks for system safety. A single star (*) indicates the interval cannot be escalated.
- **SEMR**: System Equipment Maintenance Requirement from ALS Part 4.
- **FAL**: Fuel Airworthiness Limitation from ALS Part 5, related to fuel tank safety.
- **CPCP**: Part of the Corrosion Prevention and Control Program.
- **CMP**: From the ETOPS CMP document, applicable for extended-range operations.

Reference. Provides references to other technical publications, most commonly the Aircraft Maintenance Manual (AMM) task reference (AMTOSS) for detailed procedures. It also links to the original MRBR or ALS task number.

Men / MH. These columns provide planning estimates for the minimum number of **Men** (personnel) and the **MH** (Man-Hours) required. Man-hours are broken down into time for the task itself, for access (opening/closing panels), and for preparation.

Applicability. Defines the specific aircraft to which the task applies, based on model (A318, A319, A320, A321), engine type, modification status (PRE/POST), or specific Manufacturer Serial Numbers (MSN).

B Procurement Data Structure and Helpers

This appendix provides the detailed data structures and helper algorithms that underpin the procurement model described in Chapter 4. While the main chapter focuses on the strategic logic, the components detailed here are essential for the model's implementation and day-to-day operation. They define how the model represents its knowledge of the present and future, and the mechanical steps it takes to update its beliefs as new information becomes available.

Data Structures

Table B.1: Data Structure Definitions

Variable	**Description**	**Example / Structure**
$I_d^{m,\text{fore}}$	Master forecast set of possible future inventory distributions for maintenance task m at day d. Contains $\{I_{d,d+1}^{m,\text{fore}}, I_{d,d+2}^{m,\text{fore}}, \ldots, I_{d,D_{\text{end}}}^{m,\text{fore}}\}$. This is distinct from the actual inventory level i_d^m.	$I_d^{m,\text{fore}} = [I_{d,d+1}^{m,\text{fore}}, I_{d,d+2}^{m,\text{fore}}, \ldots, I_{d,D_{\text{end}}}^{m,\text{fore}}]$
$i_{d,\tau}^{m,\text{fore}}$	Forecast distribution of possible inventory levels at future day τ based on information at day d. Each element is a pair (q_k, p_k) with q_k a possible inventory and p_k the associated probability. Linked to $i_d^{m,\text{fore}}$ through its index τ.	$i_{d,\tau}^{m,\text{fore}} = \{(q_1, p_1), \ldots, (q_K, p_K)\}, \quad \sum_{k=1}^K p_k = 1$ Example: $\{(5, 0.25), (6, 0.50), (7, 0.25)\}$
$A_d^{m,\text{expected}}$	Master set of open arrivals packets for task m known at day d: $A_d^{m,\text{expected}} = \{\mathcal{A}_{d,l}^m : l \in \mathcal{L}_{m,d}\}$. Each packet keeps arrivals separated by order day (and batch).	$A_d^{m,\text{expected}} = [\mathcal{A}_{d,l_1}^m,\ \mathcal{A}_{d,l_2}^m, \ldots]$
$\mathcal{A}_{d,l}^m$	Arrivals packet for orders of m placed on day l and still open at day d. Stores, for a regular (reg) order of fixed size $q^m l$, the conditional pmf $(q_l^m, \pi_l^m(\tau \mid d)) : \tau > d$ with $\sum_{\tau>d} \pi_l^m(\tau \mid d) = 1$. If an express order of size $q_l^{m,\text{expr}}$ exists and $l+1 > d$, include the deterministic atom $(q_l^{m,\text{expr}}, 1)$ at $\tau = l+1$.	For current day $d = 39$: A **reg** placed on $l = 32$ for a quantity of 3 could be represented by the packet $\mathcal{A}_{39,32}^m = \{(3, 0.5)_{\tau=40},\ (3, 0.3)_{\tau=41},\ (3, 0.2)_{\tau=42}\}$. An **expr** placed on $l = 39$ for a quantity of 2 would be $\mathcal{A}_{39,39}^m = \{(2, 1.0)_{\tau=40}\}$.
$\alpha_{m,l,\tau,d}^{\text{expected}}$	Forecast arrival atom for a reg order of fixed size q_l^m placed on day l, arriving on day τ, as of d, with conditional probability $\pi_l^m(\tau \mid d)$ (with $\sum \tau > d \pi_l^m(\tau \mid d) = 1$). For express, arrival is deterministic at $\tau = l+1$.	Example: $\alpha_{m,l,40,d}^{\text{expected}} = (x_{m,l}^{\text{reg}}, 0.6)$, $\alpha_{m,l,41,d}^{\text{expected}} = (x_{m,l}^{\text{reg}}, 0.4)$, and $\alpha_{m,l,l+1}^{\text{expr}} = (x_{m,l}^{\text{expr}}, 1)$.
$\mathcal{E}_\tau^m$	Expected demand pmf on day τ for task m: a set of pairs $(e,\ p_{m,\tau}^E(e))$ with $\sum_e p_{m,\tau}^E(e) = 1$.	Example: $\mathcal{E}_{40}^m = \{(0, 0.20), (1, 0.50), (2, 0.25), (3, 0.05)\}$. Then $\bar{e}_{m,40} = \sum_e e\, p_{m,40}^E(e) = 1.15$.

The model's ability to make forward-looking decisions relies on a set of carefully designed data structures, defined in Table B.1. These structures are not merely static parameters; they represent the dynamic state of the system. Key among them are $A_d^{m,\text{expected}}$, which tracks the probabilistic arrival dates of all open orders, and $I_d^{m,\text{fore}}$, which holds the complete probability distribution of future inventory levels. Together, these structures provide a comprehensive, time-aware snapshot of the system's current state and its projected evolution.

Helper Algorithms

The following algorithms perform the essential daily tasks of state updating and forecasting.

Updating Expected Arrivals

Algorithm 5 is the core ”bookkeeping” routine that runs at the start of each day. Its job is to reconcile the model's prior beliefs about future arrivals with the reality of what has just occurred. The algorithm processes any realized arrivals (regular, express, or pre-procured), removing completed orders from the open orders set $A^{\text{expected}}_{m,d-1}$. Critically, for stochastic lead times, if an order did *not* arrive on a day it was expected to, the algorithm performs a conditional probability update (renormalization) on the remaining possible arrival dates. Finally, it incorporates any new orders placed on day d, creating the updated state $A^{\text{expected}}_{m,d}$ that serves as the basis for all of the day's decisions.

Forecasting Future Inventory

Algorithm 6 is the model's short-term forecasting engine. Given the current state (on-hand inventory and the updated open orders set), it generates a complete scenario tree of all possible inventory levels for each day in the short-term horizon, $[d+1, T]$. It does this by systematically expanding every possible combination of stochastic events, both arrivals from open orders and potential demand on maintenance days, and calculating the precise probability of each resulting inventory state. This full enumeration provides the high-fidelity inventory distributions ($I^{m,\text{fore}}_{d,\tau}$) required for accurate short-term cost calculations. For the long-term tail, where such an expansion is computationally infeasible, the algorithm transitions to a simpler moment-based projection, propagating the mean and variance forward in time.

Algorithm 5. **Update of Expected Arrivals $A^{\text{fore}}_{m,d}$ given observations $\alpha^{\text{reg}}_{m,l,d}$ and $\alpha^{\text{expr}}_{m,d-1,d}$**

```
Input: $A^{\text{expected}}_{m,d-1}$, observed regular arrivals $\alpha^{\text{reg}}_{m,l,d}$, observed express arrival $\alpha^{\text{expr}}_{m,d-1,d}$, observed
       pre-procurement $\alpha^{\text{pre}}_{m,d}$, new orders $x^{\text{reg}}_{m,d}$, $x^{\text{expr}}_{m,d}$
Output: $A^{\text{expected}}_{m,d}$
$A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d-1}$
// (1) Realized regular arrivals
foreach $l \le d-1$ with $\alpha^{\text{reg}}_{m,l,d} > 0$ do
    $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \setminus \{\alpha^{\text{expected}}_{m,l,\tau} : \tau \ge d\}$
// (1') No regular arrival but positive mass at $\tau = d$: condition on survival and renormalize
foreach $l \le d-1$ with $\alpha^{\text{reg}}_{m,l,d} = 0$ and $\exists \tau \ge d$ s.t. $\alpha^{expected}_{m,l,\tau} \in A^{expected}_{m,d}$ do
    let $s^{(d)}_{m,l} \leftarrow \begin{cases} p^{(d-1)}_{m,l,d}, & \text{if } \alpha^{\text{expected}}_{m,l,d} \text{ exists} \\ 0, & \text{otherwise} \end{cases}$
    delete $\alpha^{\text{expected}}_{m,l,d}$ from $A^{\text{expected}}_{m,d}$ if it exists
    foreach $\tau > d$ with $\alpha^{expected}_{m,l,\tau} = (q^m_l,\ p^{(d-1)}_{m,l,\tau})$ do
        set $p^{(d)}_{m,l,\tau} \leftarrow \dfrac{p^{(d-1)}_{m,l,\tau}}{1 - s^{(d)}_{m,l}}$
        replace $\alpha^{\text{expected}}_{m,l,\tau}$ by $(q^m_l,\ p^{(d)}_{m,l,\tau})$
    // $\sum_{\tau>d} p^{(d)}_{m,l,\tau} = 1$
// (2) Realized express arrival (deterministic $L = 1$)
if $\alpha^{\text{expr}}_{m,d-1,d} > 0$ then
    $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \setminus \{\alpha^{\text{expr}}_{m,d-1,d}\}$
// (3) Realized pre-procurement at day $d$
if $\alpha^{\text{pre}}_{m,d} > 0$ then
    $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \setminus \{\alpha^{\text{pre}}_{m,\cdot,d}\}$
// (4) New orders placed at $d$: add to $A^{expected}_{m,d}$
if $x^{\text{reg}}_{m,d} > 0$ then
    add $\{\alpha^{\text{expected}}_{m,d,\tau} = (x^{\text{reg}}_{m,d},\ \pi^m_d(\tau)) : \tau \ge d+1\}$ to $A^{\text{expected}}_{m,d}$
if $x^{\text{expr}}_{m,d} > 0$ then
    add $\{\alpha^{\text{expr}}_{m,d,d+1} = (x^{\text{expr}}_{m,d},\ 1)\}$ to $A^{\text{expected}}_{m,d}$
return $A^{expected}_{m,d}$
```

Algorithm 6. **Future Tracking Inventory Expectation via Master Set $\mathcal{F}$**

```
   Input: current day $d$, horizon end $T$, part $m$ on-hand $i_d^m$, baseline expected-arrivals $A_d^{m,\text{expected}}$, demand
          pmfs $\{\mathcal{E}_\tau^m\}_{\tau=d+1}^T$
   Output: Master set $\mathcal{F} = \bigcup_{\tau=d+1}^T \mathcal{F}_\tau$
   Marginals $\{I_{d,\tau}^{m,\text{fore}}\}_{\tau=d+1}^T$ and moments $\{\mathbb{E}[I_\tau^m], \mathrm{Var}(I_\tau^m)\}$
   // Short Term
1  $\mathcal{F} \leftarrow \varnothing$ for $\tau \in \{d{+}1, \dots, T\}$ do
       // Initialize containers and an index counter for today's scenarios at day $\tau$
2      $\mathcal{F}_\tau \leftarrow \varnothing$ $\mathcal{K}_\tau \leftarrow \varnothing$ $\kappa_\tau \leftarrow 0$ $\mathrm{PredMap}_\tau \leftarrow \varnothing$ $\mathrm{Prob}_\tau \leftarrow \varnothing$ $\mathrm{InvEnd}_\tau \leftarrow \varnothing$ $\mathrm{ArrExp}_\tau \leftarrow \varnothing$
       // Iterate over all predecessor scenarios from day $\tau{-}1$
3      for $k' \in \mathcal{K}_{\tau-1}$ do
           // Carry predecessor's state
4          $i \leftarrow \hat{i}_{\tau-1,k'}^m$ $A \leftarrow \widehat{A}_{\tau-1,k'}^{m,\text{expected}}$
           // Adjust for certain arrivals
5          $D \leftarrow \{\, \alpha_{m,l,\tau}^{\text{expected}} = (q_l^m, 1) \ \in \ A \,\}$     $i \leftarrow i \ + \ \sum_{(q_l^m,1)\in D} q_l^m$     $A \leftarrow A \setminus D$
           // Define today's active arrivals (strictly stochastic at day $\tau$)
6          $\mathcal{A}_\tau^{\text{act}} = \{\, (l,\ q_l^m,\ \rho_{m,l,\tau}) \ : \ \alpha_{m,l,\tau}^{\text{expected}} = (q_l^m, \rho_{m,l,\tau}) \in A,\ \ 0 < \rho_{m,l,\tau} < 1 \,\}$
7          if $\mathcal{A}_\tau^{act} = \varnothing$ and $\mathcal{E}_\tau^m = \varnothing$ then
8              $\mathcal{F}_\tau \leftarrow \mathcal{F}_\tau \cup \{\, ((\tau,k),\ (\tau{-}1,k'),\ i,\ A,\ p_{\tau-1,k'}) \,\}$
9          else if $\mathcal{A}_\tau^{act} \neq \varnothing$ and $\mathcal{E}_\tau^m = \varnothing$ then
               // iterate all arrival/non-arrival combinations
10             $(\ell_1, \dots, \ell_n)$:=ordered distinct $l$ in $\mathcal{A}_\tau^{\text{act}}$, $\rho_r := \rho_{m,\ell_r,\tau}$, $q_r := q_{\ell_r}^m$
11             for $\mathbf{b} = (b_1, \dots, b_n) \in \{0,1\}^n$ do
12                 $p_\tau(\mathbf{b}) = \prod_{r=1}^n \rho_r^{b_r} (1-\rho_r)^{1-b_r}$,
13                 $L_{\text{arr}}(\mathbf{b}) = \{\ell_r : b_r = 1\}$,   $L_{\text{surv}}(\mathbf{b}) = \{\ell_r : b_r = 0\}$
14                 $i^{(\mathbf{b})} \leftarrow i + \sum_{\ell \in L_{\text{arr}}(\mathbf{b})} q_\ell^m$
15                 $A^{(\mathbf{b})} \leftarrow \Big( A \setminus \{\alpha_{m,\ell,u}^{\text{expected}} : \ell \in L_{\text{arr}}(\mathbf{b}),\ u \geq \tau\} \Big) \setminus \{\alpha_{m,\ell,u}^{\text{expected}} : \ell \in L_{\text{surv}}(\mathbf{b}),\ u >$
                    $\tau\} \ \cup \ \bigcup_{\ell \in L_{\text{surv}}(\mathbf{b})} \{(q_\ell^m,\ \hat{p}_{m,\ell,u}) : u > \tau\}$
                    where $\hat{p}_{m,\ell,u} := \dfrac{p_{m,\ell,u}}{1-\rho_{m,\ell,\tau}}$ for $u > \tau$, and $p_{m,\ell,u}$ := old prob in $A$
16                 $p^{(\mathbf{b})} \leftarrow p_{\tau-1,k'} \cdot p_\tau(\mathbf{b})$
17                 $\mathcal{F}_\tau \leftarrow \mathcal{F}_\tau \cup \Big\{ \big((\tau,k),\ (\tau{-}1,k'),\ i^{(\mathbf{b})},\ A^{(\mathbf{b})},\ p^{(\mathbf{b})}\big) \Big\}$          // save new scenario
18         else if $\mathcal{A}_\tau^{act} = \varnothing$ and $\mathcal{E}_\tau^m \neq \varnothing$ then
19             for $(e, p_e) \in \mathcal{E}_\tau^m$ do
20                 $i^{(e)} \leftarrow \max\{\, i - e,\ 0 \,\}$
21                 $\mathcal{F}_\tau \leftarrow \mathcal{F}_\tau \cup \Big\{ \big((\tau,k),\ (\tau{-}1,k'),\ i^{(e)},\ A,\ p_e\big) \Big\}$
22         else
23             $(\ell_1, \dots, \ell_n)$:=ordered distinct $l$ in $\mathcal{A}_\tau^{\text{act}}$, $\rho_r := \rho_{m,\ell_r,\tau}$, $q_r := q_{\ell_r}^m$ for $\mathbf{b} = (b_1, \dots, b_n) \in \{0,1\}^n$
                do
                   // Repeat lines 15-19 get $p_\tau(\mathbf{b})$, $L_{arr}(\mathbf{b})$, $L_{surv}(\mathbf{b})$, $i^{(\mathbf{b})}$, $A^{(\mathbf{b})}$, $p^{(\mathbf{b})}$
24                 for $(e, p_e) \in \mathcal{E}_\tau^m$ do
25                     $i^{(\mathbf{b},e)} \leftarrow \max\{\, i^{(\mathbf{b})} - e,\ 0 \,\}$ $\mathcal{F}_\tau \leftarrow \mathcal{F}_\tau \cup \Big\{ \big((\tau,k),\ (\tau{-}1,k'),\ i^{(\mathbf{b},e)},\ A^{(\mathbf{b})},\ p^{(\mathbf{b})}\, p_e\big) \Big\}$
   // Assemble Master Set
26 $\mathcal{F} \leftarrow \bigcup_{\tau=d+1}^T \mathcal{F}_\tau$ for $\tau \in \{d{+}1, \dots, T\}$ do
27     $I_{d,\tau}^{m,\text{fore}} \leftarrow \{(\hat{i}_{\tau,k}^m, p_{\tau,k}) : k \in \mathcal{K}_\tau\}$          // Can Compute $\mathbb{E}[I_\tau^m]$ and $\mathrm{Var}(I_\tau^m)$ from $I_{d,\tau}^{m,fore}$
   // Long Term
28 $\mathbb{E}[I_T^m] \leftarrow \sum_{k \in \mathcal{K}_T} \hat{i}_{T,k}^m\, p_{T,k}$,   $\mathrm{Var}[I_T^m] \leftarrow \sum_{k \in \mathcal{K}_T} \big(\hat{i}_{T,k}^m - \mathbb{E}[I_T^m]\big)^2 p_{T,k}$
29 for $\tau \in \{T{+}1, \dots, D\}$ do
30     $\bar{e}_{m,\tau} \leftarrow \sum_e e\, p_{m,\tau}^E(e)$ (else 0),   $\sigma_{D,m,\tau}^2 \leftarrow \sum_e (e - \bar{e}_{m,\tau})^2 p_{m,\tau}^E(e)$ (else 0)
31     $\mathbb{E}[I_\tau^m] \leftarrow \max\{\, \mathbb{E}[I_{\tau-1}^m] + a_{m,\tau}^{\text{pre}} - \bar{e}_{m,\tau},\ 0 \,\}$,   $\mathrm{Var}[I_\tau^m] \leftarrow \mathrm{Var}[I_{\tau-1}^m] + \sigma_{D,m,\tau}^2$
32 return $\mathcal{F}$, $\{I_{d,\tau}^{m,fore}\}$, $\{\mathbb{E}[I_\tau^m], \mathrm{Var}(I_\tau^m)\}$
```

C Scientific Paper

Aircraft Maintenance Repair & Overhaul Spare Parts Management

Christos Paschalidis

Managing spare parts for Aircraft Maintenance, Repair, and Overhaul (MRO) is challenging because there is a significant gap between long-term maintenance schedules and daily procurement decisions. While existing research often addresses demand forecasting and inventory control in isolation, this paper introduces a complete framework to bridge this gap by directly connecting day-to-day procurement decisions with the fixed, fleet-wide maintenance schedule. The framework employs a task-based approach to enhance traditional planning. It consists of two modular stages. First, a demand forecasting methodology converts raw maintenance tasks into a time-phased, probabilistic demand signal by systematically grouping them based on their technical attributes. Second, a daily procurement optimization model acts on this detailed forecast. It replicates a planner's decision-making by explicitly comparing the expected future costs of buying, waiting, or selling surplus stock. This creates a transparent, cost-based model where every decision accounts for probabilistic predictions and remains fully auditable. When validated against a conservative strategy and a periodic-review benchmark, the proposed model reduced total net costs for both, achieving savings of 17.5% and 9.2% respectively. The analysis shows this advantage stems from the strategic acceptance of controlled risks when doing so leads to a lower expected total cost.

I. Introduction

Aircraft maintenance planning has improved greatly in recent years, becoming a much more organized process. This change is mainly driven by strict regulations where safety and reliability are top priorities. To meet these standards, operators must carefully track all maintenance tasks, paying attention to their specific timing and how critical they are. A part of this organized approach is the Maintenance Planning Document (MPD), which manufacturers provide as a central document for all repeating tasks. This core document gathers requirements from various official sources, like the Maintenance Review Board Report (MRBR) and Airworthiness Limitation Sections (ALS), into a single set of Instructions for Continued Airworthiness (ICA). This centralized document provides the detailed, long-term schedule that forms the backbone of an airline's maintenance program.

Spare parts provisioning is often guided by International Air Transport Association (IATA) classifications, which segment inventory based on value, repairability, and financial treatment. Operators typically distinguish between high-value, depreciable **Rotables** (e.g., wheels); medium-value **Repairables** (e.g., starter motors) which are described by scrap rates; and single-use **Expendables** (e.g., filters, seals). Beyond these standard classifications, the state-of-the-art has advanced into multi-criteria ranking and reliability-based forecasting. Research has also explored integrated models that link maintenance plans with inventory needs to optimize costs. Despite these theoretical advances, practical procurement strategies still often rely on aggregate historical data. This aggregate-level planning is fundamentally incompatible with the rising adoption of Equalized Maintenance Programs (EMPs). EMPs distribute heavy maintenance into smaller, frequent checks to maximize aircraft availability, a methodology that demands a precise, time-phased flow of materials that traditional methods cannot guarantee.

This paper investigates a complete framework designed to connect the detailed, fleet-wide maintenance schedule directly to inventory decisions. This framework functions as a pipeline, taking the established list of scheduled tasks as its input and producing calendar-dated procurement actions as its output. It consists of two primary components: a demand forecasting methodology and a procurement optimization model. The forecast moves beyond aggregate data by deriving replacement probabilities from the specific technical attributes of each maintenance task. This framework does not track individual components; instead, it uses a simplification where each maintenance task is linked to a single type of spare part. This task-specific, probabilistic demand signal then feeds into a cost-driven procurement strategy, which determines the optimal ordering policy.

The paper is organized as follows. Section II reviews the relevant literature on spare parts management. Section III details the complete methodology, first explaining the task-based demand forecasting and second, the cost-driven procurement model. Section IV presents the results, where the framework is benchmarked and analyzed. Section V discusses the operational utility and strategic implications of the framework. Finally, Section VI offers concluding remarks.

II. Related Work

The spare parts management literature relevant to this work falls into two main streams of research. The first, and largest, stream focuses on optimizing standalone elements of the procurement process. This includes dedicated studies on parts classification, demand forecasting for intermittent parts, and reliability modeling. The second stream explores integrated approaches, where models are built to connect two or more of these functions, such as coupling demand forecasting directly with inventory control.

Traditional spare parts classification methods, common across many industries, are also applied in aviation. Single-factor classifications like ABC analysis, however, often fail to capture the diverse drivers of MRO inventory management (Roda et al., 2014). Qualitative schemes such as VED (Vital, Essential, Desirable) are also used, but they rely on experience and lack reproducible, measurable rules. Consequently, current research emphasizes multi-criteria frameworks that simultaneously consider part criticality, demand frequency, lead time, and cost (Roda et al., 2014). Despite this, a gap often exists between advanced academic models and the simplistic methods used in practice (Bacchetti and Saccani, 2012). Recent studies, such as the AHP-based model developed by Ayu Nariswari et al. (2019), have sought to provide more transparent, multi-criteria prioritization for MROs.

Classical base stock rules are often built around a single demand distribution. For situations where demand drifts or spikes, alternative approaches have been developed. Kang et al. (2023) frame inventory planning as an adaptive robust optimisation problem where demand can follow any path within a predefined uncertainty set; the policy is sized to maintain service targets even under the most adverse path. This robust policy was reported to match fill rates while tying up much less capital than a Poisson-based benchmark. Other tactics use fixed thresholds on key inventory signals. Hekimoğlu et al. (2022), for instance, present a monitoring scheme for repairable parts that routes units through an expedited, higher-cost repair channel when the in-repair queue passes a critical threshold, cushioning downtime more economically than large safety stocks.

Recent research optimizes maintenance scheduling and spare provisioning jointly, rather than as separate problems. Qin et al. (2020) use a scenario-based stochastic programming model to integrate maintenance plans with inventory decisions. Their two-stage approach decides on rotable overhaul modes and pre-purchasing upfront, then uses emergency procurement after uncertain demand is realized, minimizing total costs. Integration also extends to resource capacity. Erkoc and Ertogral (2016) tackle the scheduling of rotable overhauls constrained by both limited spares and finite repair capacity, optimizing the schedule to minimize premature component removals. In a similar sense, Buyukkaramikli et al. (2015) coordinate repair capacity investment with inventory control, finding the optimal balance between holding more spares and expanding shop throughput.

The literature provides a strong foundation, offering effective tools for parts classification, demand forecasting, and robust inventory control. Researchers have also demonstrated the benefits of linking maintenance planning with inventory decisions. However, this body of work predominantly focuses on optimizing static inventory policies, such as safety stock levels, rather than guiding the specific, daily decisions a planner must make. A clear gap exists for a framework that explicitly associates real-world maintenance tasks with their specific demand probabilities to drive decision support. This missing link prevents the direct translation of a fleet-wide maintenance schedule into auditable, calendar-dated procurement orders that are steered by operational performance metrics.

This leads to the central research question guiding this work:

> *RQ: How can a maintenance plan be translated into a procurement strategy that minimizes cost by leveraging task-specific demand forecasts?*

III. Methodology

To address the central research question, the framework's design is defined by a set of core principles. These principles distinguish the proposed approach from traditional aggregate methods:

- **Calendar Fidelity:** Decisions are made on the same daily timeline as the maintenance plan, utilizing the most current information rather than relying on abstract, periodic reviews.
- **Task-Level Granularity:** The focus is placed entirely on the replacement potential of individual, scheduled tasks, abandoning aggregate-level forecasts.
- **Cost Traceability:** Every procurement recommendation must be fully auditable. The financial trade-offs between purchasing, holding inventory, and risking a stockout are rendered transparent and explainable.

Guided by these principles, the methodology is built as a sequential pipeline organized into two distinct stages. First, the **Demand Forecasting** stage (Section A) details the process of translating the raw maintenance plan into a task-specific, probabilistic demand signal. Following this, the **Procurement Model** (Section B) is introduced, describing how this detailed forecast is utilized to generate daily, cost-based ordering decisions.

A. Demand Forecasting

The first part of the methodology, outlined in Figure 1, is to generate this demand signal. This involves a process that starts with the raw task descriptions in the MPD, systematically groups them based on their technical attributes, and ends by applying a reliability model to a simulated maintenance schedule.

1. Data

Effective procurement depends on a reliable demand forecast. When historical maintenance records are unavailable, such as during 'cold starts' (new fleets or operators), a forecast must be synthetically generated. This research generates a synthetic, logically-grounded demand signal for the procurement model. The methodology is modular, allowing refinement as operational data becomes available. The process begins using the Maintenance Planning Document (MPD), the manufacturer's central document for repetitive maintenance tasks, which provides structured data like task intervals and zones. However, critical information for failure potential is often in the unstructured, free-text "Description" field. To convert this, a Large Language Model (LLM) is employed not as a predictor, but as a feature engineering tool. The LLM processes each task's description and, guided by a specific prompt, assigns a set of new, structured categorical attributes.

The feature engineering process results in a final set of eight categorical attributes. Four are derived directly from the structured MPD data, while the remaining four are synthesized by the LLM to capture unstructured technical context. These attributes, summarized in Table 1, form the input vector for the clustering algorithm.

Although an expert-validated reference is not available to mathematically prove the LLM's absolute accuracy, the consistency of the method is verifiable and essential for a repeatable framework. The enrichment process was run four additional times on the same MPD data, demonstrating high stability. The assignments for 'Outcome Expectation' were the most consistent (98.8% identical across all runs), while 'Inspection Access Level' was the least (89.1% identical). This confirms the LLM provides a stable foundation for the analysis. These new attributes also quantify the nature of the maintenance work in ways the original data does not; for example, the results show that most tasks relate to 'Static' components (70.9%) and require non-intrusive access ('External Surface Only' or 'Limited Internal Access' totalling 86.1% of tasks).

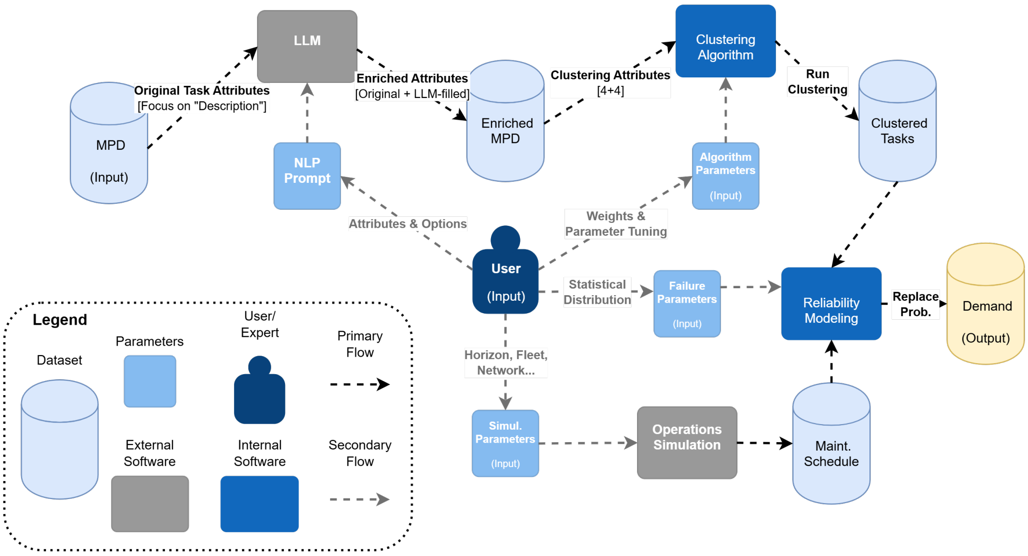


Figure 1. Process Flow for Generating a Synthetic Spare Part Demand Forecast.

Table 1. Categorical Attributes Used for Clustering

Attribute	Description and examples
Generated by LLM	
Exp. Failure Mode	Dominant physical mechanism (e.g., structural, electrical).
Comp. Movement	Motion status during operation (e.g., static, moving).
Insp. Access Level	Depth of access required (e.g., surface only, disassembly).
Outcome Exp.	Expected end state (e.g., visual continue, replace).
Derived from MPD	
Source	Regulatory origin of the task (e.g., MRB, ALS).
Task Code	Type of maintenance work (e.g., inspection, servicing).
Zone Count	Number of aircraft zones affected (e.g., 1, 2, 3+).
Frequency	Occurrence rate over horizon (e.g., <1 time, 10+ times).

2. Clustering

The eight categorical attributes are used to cluster the maintenance tasks. The objective is to group tasks that are expected to share a similar relationship between the maintenance action and the potential need for a spare part, effectively creating reliability groups. This method provides a practical way to manage the large, diverse set of tasks. It also establishes a baseline for assigning new tasks to a group when their specific replacement behaviour is unknown. While these clusters could ideally be validated or tuned using historical failure data or expert consultation, this framework provides a starting point in the absence of such information.

Because all eight attributes are categorical, a suitable clustering algorithm must be able to handle non-numeric data, rendering standard distance-based methods like K-means ineffective. Several specialized algorithms exist for this purpose, such as K-modes, Latent Class Analysis (LCA), and Hierarchical Agglomerative Clustering (HAC). The optimal choice would ideally be validated by matching cluster outputs to historical failure patterns or through expert consultation. For this study, the ROCK (RObust Clustering using linKs) algorithm is employed. By utilizing a link-based similarity measure rather than simple distances, this approach effectively captures the structural context of the data, yielding interpretable and relatively uniform clusters that serve as a robust foundation for reliability modelling.

ROCK is a hierarchical clustering algorithm for categorical data that measures similarity using shared neighbors, or "links," rather than direct distances. First, a similarity metric (in this study, the Jaccard coefficient) is used to identify "neighbors" for each task t_i: two tasks are neighbors if their similarity exceeds a given threshold θ. The algorithm then defines the number of links between two tasks, $\text{link}(t_i, t_j)$, as the total count of common neighbors they share. ROCK builds clusters from the bottom up by iteratively merging the pair of clusters (C_i, C_j) that maximizes a goodness measure, $g(C_i, C_j)$. This measure promotes high internal link density and is defined as:

$$g(C_i, C_j) = \frac{\text{link}[C_i, C_j]}{(n_i + n_j)^\epsilon - n_i^\epsilon - n_j^\epsilon}$$

where n_i is the size of cluster C_i, $\text{link}[C_i, C_j]$ is the total links between the clusters, and ϵ is an exponent that normalizes for cluster size. The complete logic is detailed in Appendix A.

The process first segregates all mandatory replacement tasks (e.g., Discard, Restore) into a dedicated Cluster 1, as these always require a part. The ROCK algorithm is then applied to the remaining tasks, partitioning them into seven additional clusters. A total of eight groups offers a practical trade-off, providing enough distinct categories to capture various failure behaviors while avoiding the fragmentation of creating too many small clusters. The resulting groups are thematically coherent, with each defined by a dominant profile. For example, a large "Structural Inspection" cluster emerged, characterized by a high concentration of the 'Structural' failure mode and the 'Visual Structural Continue' outcome. The complete profiles for all clusters are detailed in Appendix A.

3. Reliability Modelling

Following clustering, each task group is assigned a reliability model. This model is a statistical distribution that describes the probability that performing a maintenance task will result in a spare part replacement. The nature of this replacement demand depends on the part category. For Expendables a faulty part is always scrapped, creating a direct demand for a new unit. For Repairables like starter motors, a faulty part may either be repaired, sometimes on-aircraft and sometimes in a shop, or be deemed beyond economic repair and scrapped, which also creates a demand for a replacement.

For this study, the two-parameter Weibull distribution is selected to model the probability of a component replacement. This model is defined by a scale parameter, α, and a shape parameter, β. The shape parameter β is particularly descriptive, as $\beta < 1$ suggests a decreasing failure rate (infant mortality), $\beta = 1$ indicates a constant rate (random failures), and $\beta > 1$ points to an increasing wear-out failure rate. The primary function used is the Cumulative Distribution Function (CDF), $F(t)$, which gives the probability that a component fails at or before age t:

$$F(t) = 1 - \exp\left[-\left(\frac{t}{\alpha}\right)^\beta\right], \quad t \geq 0$$

The model's Reliability Function, $R(t)$, or the probability of surviving beyond age t, is the complement of the CDF:

$$R(t) = \exp\left[-\left(\frac{t}{\alpha}\right)^\beta\right]$$

Finally, the Failure Rate Function, $h(t)$, gives the instantaneous risk of failure at age t given survival up to that point:

$$h(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}$$

To apply this model, the parameters are directly linked to the maintenance plan. The **scale parameter** α is set as the task's scheduled maintenance interval, measured in Flight Hours, Flight Cycles, or Days. The **time variable** t is the component's age at the time of the check, using the same unit. As no historical data is available to estimate the **shape parameter** β, values are assigned incrementally, ranging from $\beta = 1.1$ for Cluster 2 up to $\beta = 1.7$ for Cluster 8. This reflects an assumption of various wear-out failure processes to create diversified reliability groups, not an empirically validated model. This Weibull analysis is applied to all task clusters except for Cluster 1; its mandatory replacement tasks are excluded from this calculation and are assigned a fixed failure probability of 1.0.

4. Application

The framework applies this reliability modeling to an operations simulation of a fleet of eight A320-family aircraft over a five-year planning horizon. This simulation employs an equalized work package strategy, which differs from traditional, heavy letter checks. This approach groups maintenance tasks into small, frequent blocks designed to be completed within short, overnight ground slots, allowing for normal flight operations during the day. Unlike heavy checks which provide multi-day buffers for logistics, these restricted overnight slots preclude reactive procurement, requiring immediate part availability. Based on the detailed flight and maintenance schedule generated, the age of each component is tracked. When a task becomes due, the component's current age is used as the input t for its cluster's Weibull model, calculating the specific, calendar-dated probability that this maintenance instance will lead to a part replacement. Figure 2 illustrates this output, showing the dynamic failure probabilities for two sample tasks as their age fluctuates, alongside a third task shown with a mandatory replacement.

This process is repeated for every scheduled task across the entire five-year planning horizon, producing the final output of this chapter: a complete, time-phased demand forecast. This forecast is highly granular, providing a replacement probability for each specific maintenance instance. When these individual probabilities are aggregated, the total expected monthly demand can be visualized, as shown in Figure 3. The resulting demand profile is highly volatile and non-steady, with sharp peaks in resource needs followed by quiet months. This uneven demand is a direct consequence of how tasks are bundled in the equalized schedule. Such volatility makes it difficult for a human planner to rely on experience or standard rules of thumb, as the specific needs of one period rarely predict the next. The procurement and inventory optimization model, detailed in the next section, is tasked with managing this fluctuating, task-specific demand signal.

B. Procurement Model

The time-phased, probabilistic demand forecast generated from the maintenance plan provides the core input for the procurement optimization model. This model is designed to act on this granular, task-specific signal, creating the crucial link between the fixed maintenance schedule and the uncertain supply chain. The framework tackles the core challenge of deciding what to order and when, navigating uncertainty in both demand and delivery times by operating on a daily, rolling-horizon basis.

The model tackles this as a single-item stochastic inventory control problem. Its purpose is to consolidate all the information available to a planner and use computing power to rigorously incorporate uncertainty into cost-driven trade-offs. It employs a deterministic and myopic policy, intentionally mirroring the logic a human planner might use. The policy is deterministic because it will always select the exact same action when faced with an identical state and the same future probability distributions. It is myopic because it optimizes only for the decision being made today; it does not attempt to solve for a globally optimal policy across all future decisions, as a full dynamic program would. This deliberate design choice makes the model computationally fast and, most importantly, keeps its recommendations transparent and explainable.

1. Core Elements

Initialization Before detailing the daily evaluation logic, the model's core procurement mechanisms must be defined. First, a baseline supply is established via a **pre-procurement schedule**. This is a fixed stream of deliveries, determined once at the start of the horizon by calculating a high-confidence quantity γ of the total expected demand, $Q_m^{\text{pre}}(\gamma) = \max\{\, q \in \mathbb{Z}_{\geq 0} : \Pr[X_m \geq q] \geq \gamma \,\}$, and distributing these units across fixed intervals. The daily optimization model then manages the remaining uncertainty using three distinct actions.

Actions The primary action is to place a **regular order**, which has a lower cost but an uncertain, stochastic lead time modeled as a discrete triangular distribution. As a fallback, the model can use a costly **express order**, which guarantees next-day arrival. This is an autonomous mechanism, evaluated only on the day before a maintenance task and triggered only if the expected stockout penalty exceeds the express cost. Finally, the model can **sell excess inventory** at a recovery price, which is not treated as a profit driver but as a safety valve that makes proactive procurement decisions more flexible.

Decisions are driven by a transparent balance of costs, which include the part's purchase price, regular and express shipping fees, daily holding costs, and a cancellation penalty. This penalty is the model's mechanism for pricing a service failure. If a part is required for a replacement (such as an expendable or

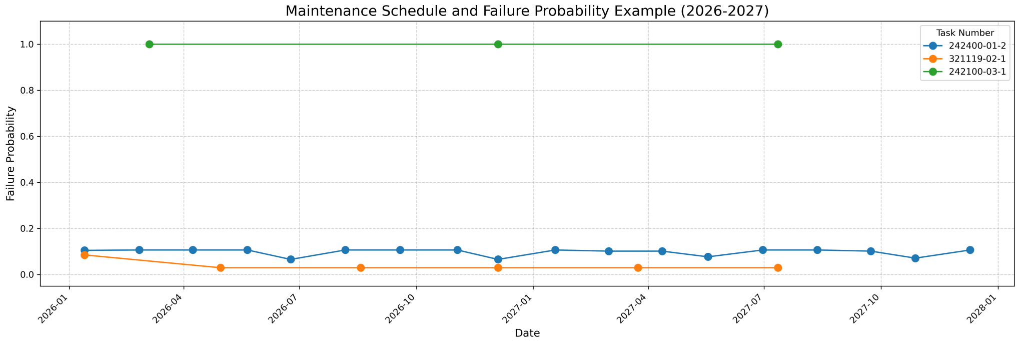


Figure 2. Maintenance Schedule and Failure Probability for Three Tasks; A318-112 (2026-2027)

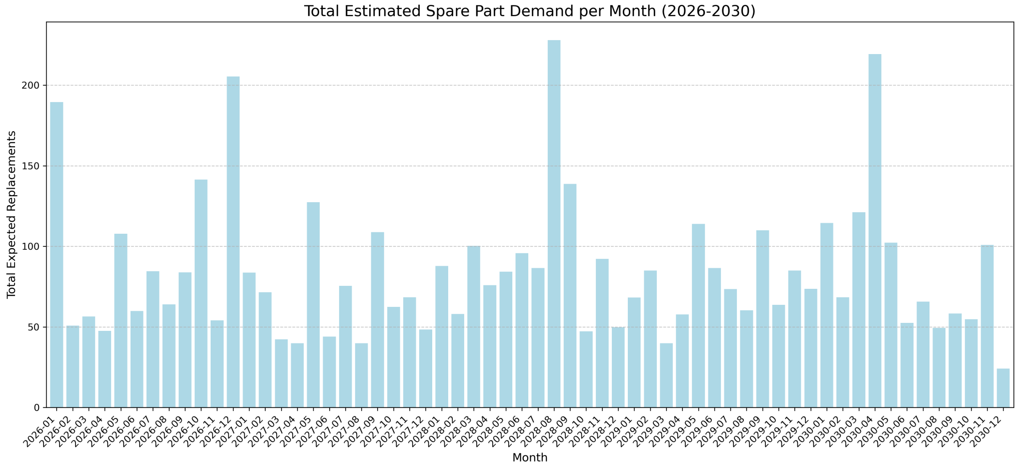


Figure 3. Total Estimated Spare Part Demand per Month (2026-2030)

a repairable part that is scrapped) and is not available in stock, the maintenance task is considered cancelled. This event incurs the significant financial penalty, which is assumed to represent the total operational consequence, such as an Aircraft on Ground (AOG). The model navigates two primary uncertainties when making these cost-based decisions: the stochastic, task-specific demand and the uncertain lead times of regular orders. All decisions are based strictly on the information available on the day the decision is made.

2. Implementation

A high-level overview of the model's daily decision process is presented in Figure 4, showing the flow from state updates to action evaluation and execution.

The model's logic operates on a daily, rolling basis. At the beginning of each day d, the system's state is updated to reflect all realized events from the previous day, including any new arrivals from open orders and any inventory consumed by maintenance tasks. To evaluate a decision, the model splits the planning horizon into two distinct periods: a short-term window $\mathcal{T}$ and a long-term tail $\mathcal{T}_{\text{LT}}$. The short-term window extends to the maximum possible lead time of a new regular order ($T = \min\{d + L_{\max}, d_{\text{end}}\}$). Within this critical window, the model performs a high-fidelity analysis by projecting the full probability distributions of future inventory, enumerating every possible combination of stochastic arrivals and demand events. Beyond this window, in the long-term tail, this full enumeration becomes computationally infeasible. The model therefore switches to a more efficient approximation, propagating only the first two moments (mean and variance) of the inventory distribution forward.

The model's master routine methodically explores the decision space by comparing the total expected future cost of three mutually exclusive actions: waiting, selling, or placing a regular order. It first evaluates the "do nothing" action to establish a baseline cost V_0. It then explores all possible sell quantities and all regular buy quantities (up to a predefined maximum, $Q_{\text{buy}}^{\max}$), using a core cost-evaluation engine to determine the expected future cost of each action. This engine calculates the total expected cost by summing purchase, shipping, and holding costs over both horizons, less any revenue from sales. A key component of this calculation is the expected shortfall cost, $\Psi(q, e)$, which is triggered on a maintenance day if demand e exceeds available inventory q:

$$\Psi(q, e) = \text{PEN}^m \cdot \mathbf{1}\{e > q\} + P^m \cdot \max\{e - q, 0\}$$

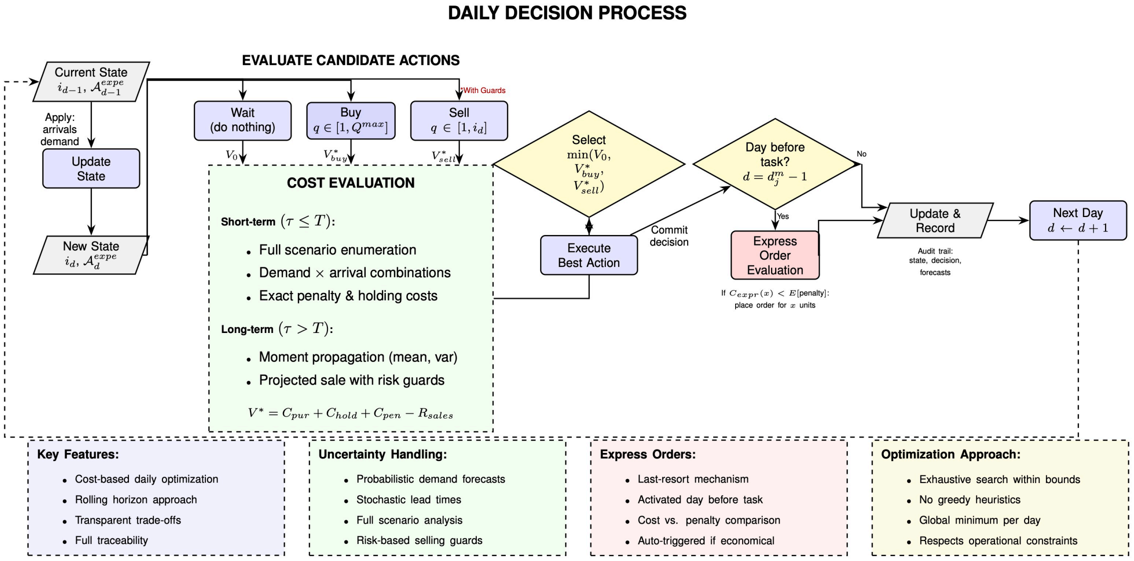
This function captures both the fixed operational penalty (PEN^m) for the cancellation and the variable cost (P^m) of the unfulfilled units.

The master routine selects the single action (wait, sell, or buy) that returns the lowest total expected cost, $V^{\text{best}} = \min\{V_0, V^{\star}_{\text{sell}}, V^{\star}_{\text{buy}}\}$. In practice, this means the model will not buy a part until the expected penalty cost of a future stockout becomes greater than the certain cost of procurement and holding. To improve long-term performance and drive the model toward a more robust strategy, a tuning lever is introduced. The model is calibrated to perceive the stockout penalty as being significantly higher than its actual financial value (e.g., 175% in this study). This "caution" buffer shifts the model's logic away from simply balancing costs, stimulating a more conservative approach that avoids penalties and finds a more effective balance between service and cost. This same cost-tradeoff logic is used independently for the autonomous express order mechanism.

Finally, the ability to sell is controlled by a 'sale guard' system. This is necessary because the model's forward-looking evaluation treats the purchase price of existing stock as a sunk cost. Without a guard, the model might sell a part to save on holding costs, only to buy it back later at a higher price, creating a financially detrimental loop. To prevent this, a sale is only considered if it passes a two-layer check. First, a minimum price threshold restricts potential sales to high-value items only (€500≤ in this study) to ensure the transaction is economically meaningful. Second, the action must pass a risk assessment. The default guard uses Cantelli's inequality, a conservative, distribution-free method that projects long-term inventory moments (μ_T, σ_T^2) and only permits a sale (leaving q units) if the worst-case probability of inventory dropping to that level is below a set risk tolerance α:

$$\frac{\sigma_T^2}{\sigma_T^2 + (\mu_T - q)^2} \leq \alpha$$

The detailed pseudocode for the core cost-evaluation engine is available in Appendix D.



Notation: i_{d-1}, i_d = inventory at end of day $d-1$ and d; $\mathcal{A}_d^{\mathrm{expe}}$ = open orders with expected arrival PMFs at day d; q = decision quantity; $Q^{\max}$ = buy cap; V_0 = expected cost if wait; V_{buy}^*, V_{sell}^* = best expected costs over feasible q; τ = day index in evaluation; T = last day of short-term window; d_j^m = maintenance date j of task m; $C_{\mathrm{expr}}(x)$ = cost of an express order of x units; E[penalty] = next-day expected stockout penalty; $V^* = C_{\mathrm{pur}} + C_{\mathrm{hold}} + C_{\mathrm{pen}} - R_{\mathrm{sales}}$.

Figure 4. Daily Procurement Decision Logic

3. Case Study

The procurement logic is driven by a set of economic and operational parameters. The specific values used in this study do not represent a single, optimal configuration but rather a baseline for analysis. The primary objective is to treat these inputs as transparent, adjustable levers. This allows for a sensitivity analysis (detailed in the next section) to test their influence and generate practical insights for a planner. The baseline configuration is summarized below.

Part Prices Since component price data was unavailable, a realistic, skewed price list was synthesized from a reference dataset on a DHC8 aircraft. This four-step process involved scaling the source data, using a weighted bootstrap to draw prices, applying a probabilistic split to categorize parts (Expendable or Repairable), and calibrating the final total monetary value.

Cost Structure All costs are dynamically derived from a part's purchase price (P). The baseline holding cost (h) is set at a 10% annual rate ($h = (0.10 \cdot P)/365$). The selling price (S) is $S = 0.80 \cdot P$. The cancellation penalty (PEN^m) is $\mathrm{PEN}^m = 200,000€ + 5 \cdot P$. Regular and express shipping costs are $C_{\mathrm{reg}} = 5€ + 0.01 \cdot P$ and $C_{\mathrm{expr}} = 50€ + 0.10 \cdot P$, respectively.

Lead Times Lead times for regular orders are stochastic, modeled as discrete triangular distributions. The baseline for Expendable parts uses a 5-day support with a mode at Day 5. Repairable parts use a 7-day support with a mode at Day 10.

IV. Results

A. Benchmark

To contextualize and evaluate the performance of the proposed cost-driven model, its results are compared against two benchmark strategies. The first is a **Fully Conservative** strategy, which simulates a highly risk-averse policy. This agent makes procurement decisions based on deterministic, worst-case assumptions, procuring enough to cover the maximum possible demand combined with the longest possible lead time, aiming for a 100% service level. The second is a **Periodic Review (12-Month Cycle)** strategy, which mimics a common "top-up" policy. This agent places a single order at the beginning of each year to raise the inventory level to a target of 120% of that year's total expected demand. For the rest of the year, it cannot place regular orders and must rely on last-minute express orders to cover any shortfalls.

The proposed model achieved the lowest Final Net Cost, outperforming the Fully Conservative strategy by 17.5% and the Periodic Review strategy by 9.2% (Table 2). The most important insight, however, comes from how it achieved this.

The metrics reveal that the model operates with a significantly leaner inventory profile. It maintained the lowest average time in stock at 252 days, compared to 300 and 406 days for the benchmarks. This efficiency drove holding costs down to €9.2 million, well below the €13.6 million and €14.5 million incurred by the alternative strategies. Furthermore, the model avoided the massive upfront spending of the conservative approach, reducing total purchase costs by over €16 million.

Counterintuitively, the proposed model incurred the highest penalty cost (€3.5 million) from 8 task cancellations, whereas the periodic-review approach had €1.5 million from 4 (fully conservative is by definition 0). This is not a failure, but evidence of the core logic succeeding: the model consistently weighs the certain cost of procurement against the expected cost of a stockout. The 8 cancellations represent the few times this calculated risk resulted in a stockout; the model's financial victory comes from the many unobserved instances where it correctly chose not to buy an unnecessary part, saving on procurement and holding costs. This flexibility contrasts sharply with the benchmarks: the Periodic model was forced into 1,577 expensive express orders, while the Conservative model paid over 50% more in holding costs to achieve perfect service.

The heavy reliance on express orders in the Periodic Review strategy reveals a critical dynamic. When the penalty for a service failure is high compared to the cost of a part, stocking for average demand is often insufficient, even with a safety buffer. The express mechanism activates so frequently because the high penalty drives the system to cover the worst-case need rather than the average one. Simply increasing the initial safety stock buffer is not an efficient fix. While it might lower shipping costs, it would drastically increase holding costs without offering much improvement on the already low cancellation rate.

B. Insight

The model's utility as a decision-support tool is demonstrated by testing its sensitivity to key parameters. The analysis isolates one variable at a time, using the default configuration from Table 2 as the baseline. The results of these simulation runs, shown in Table 3, quantify the operational and financial impact of different external risks and internal policy choices.

The analysis of external risks reveals two distinct findings. First, the model's results show that supplier reliability has a direct and significant financial impact. When the model is run with a High Uncertainty lead time scenario, the final net cost increases by 5.5%. This cost is driven by the model being forced to compensate by placing 47% more express orders, from 157 to 231 units. This increase indicates that high variance often forces the system to double-order for a single requirement. If a placed regular order fails to arrive in time due to lead time variability, the model must trigger a reactive express order to avoid a penalty. The delayed regular unit eventually arrives and becomes surplus stock, which explains the concurrent rise in holding costs.

The impact of a biased demand forecast, defined here as a systematic under- or overestimation of failure probabilities, is surprisingly small. The Final Net Cost remains stable (within 1.5%) even when the forecast is wrong by ±20%. This stability occurs for two reasons. First, the bias tested only shifted demand probabilities, not the worst-case quantity needed for a task. Because the stockout penalty is high, the optimization logic determines that procuring to cover the worst-case demand remains the lowest expected cost decision, even when that scenario's probability is underestimated. Second, the stable net cost hides a direct financial balance: the -20% run's savings on procurement are offset by a near-doubling in penalty costs (to €6.25 million). Conversely, the +20% run's savings on penalties are offset by a 15% increase in holding costs. If statistical evidence shows that demand is consistently misspecified, the failure probability parameters should be adjusted to align with operational reality.

The actual financial value of a stockout penalty dictates the trade-off between procurement spending and service failures. The analysis shows this relationship is non-linear. In an environment where the true penalty is low, the lowest final net cost (31.55M€) is reached, a 9.5% reduction from the baseline. This financial outcome occurs because it is cheaper to pay 23 cancellation penalties at this low value than to pay the high procurement and holding costs required to avoid them. Conversely, a high Penalty environment forces a near-perfect service strategy (1 cancellation) and results in a higher net cost (35.18M€).

This logic can be managed by adjusting the perceived penalty, or "caution" multiplier, which serves as a key planner assumption and is set to 1.75× in the baseline. The analysis shows that acting with no caution (a 1.0× multiplier) is not optimal, leading to 14 cancellations and a 2.8% higher net cost. A moderate level of caution (a 2.0× multiplier) identifies an operational sweet spot. This approach yields the lowest net cost (-2.2%) by finding the most effective balance between proactive spending and penalty avoidance, demonstrating that being slightly more cautious than the true cost is financially beneficial.

Table 2. Performance Comparison of Procurement Models

Performance Metric	Proposed Model	Fully Conservative	Periodic Review (12-Mo)
Service & Operational Performance			
Service Level (by days)	99.97%	100.00%	99.98%
Fill Rate (by units)	99.82%	100.00%	99.98%
Fulfilled Task Instances	4,031	4,039	4,035
Cancelled Task Instances	8	0	4
Express Orders (units)	157	0	1,577
Avg. Time in Inventory (days)	251.90	299.57	405.70
Inventory & Supply Flow (in units)			
Opening Inventory	1,074	1,074	1,074
Pre-procurement Arrivals	2,098	2,098	4,025
Repair Returns	1,119	1,119	1,119
Regular Arrivals	3,731	4,705	-
Express Arrivals	157	-	1,577
Total Inflow	**8,179**	**8,996**	**7,795**
Demand Served (Replacements)	4,404	4,407	4,406
Swaps Served (Repairs)	1,170	1,175	1,172
Units Sold	585	0	0
Closing Inventory	2,020	3,414	2,217
Total Outflow	**8,179**	**8,996**	**7,795**
Cost Breakdown (in €)			
Total Purchase Costs	34,994,006.80 (72.9%)	51,108,014.42 (77.6%)	32,673,165.00 (67.8%)
– Pre-procurement	*15,130,714.82*	*15,130,714.82*	*28,163,713.47*
– Regular	*18,352,676.05*	*35,977,299.60*	*0*
– Express	*1,510,615.93*	*0*	*4,509,451.53*
Total Shipping Costs	330,907.66 (0.7%)	270,464.12 (0.4%)	393,453.20 (0.8%)
– Regular	*186,343.30*	*270,464.12*	*0*
– Express	*144,564.36*	*0*	*393,453.20*
Holding Cost	9,174,077.14 (19.1%)	14,507,026.28 (22.0%)	13,631,597.90 (28.3%)
Penalty Cost	3,513,721.50 (7.3%)	0 (0.0%)	1,506,136.76 (3.1%)
Total Operational Costs	**48,012,713.10**	**65,885,504.82**	**48,204,352.86**
Financial Summary (in €)			
Sales Revenue	-7,461,486.29	0	0
Inv. Value Change (Gain (-)/Loss (+))	-6,078,638.51	-24,281,111.96	-10,142,085.32
Final Net Cost*	**34,855,332.60**	**40,968,521.85** (+17.5%)	**38,063,494.90** (+9.2%)

*Final Net Cost is calculated as Total Operational Costs plus/minus Inventory Value Change, less Sales Revenue. Closing inventory is valued at its sell price (80% of purchase cost).

Table 3. Sensitivity Analysis: Comparison of Key Simulation Runs

Experiment (Lever)	Scenario	Final Net Cost (M€)	Δ vs. Baseline	Cancellations	Expr. Orders	Hold. Cost (M€)
Baseline	**Default**	**34.86**	**—**	**8**	**157**	**9.17**
Lead Time	High Unc.	36.76	+5.5%	12	231	9.26
Demand Bias	-20%	34.92	+0.2%	15	200	8.18
	+20%	35.36	+1.5%	4	142	10.55
Penalty Cost	Low	31.55	-9.5%	23	180	7.97
	High	35.18	+0.9%	1	161	10.31
Perceived Pen.	1.0× (None)	35.83	+2.8%	14	158	8.47
	2.0× (High)	34.08	-2.2%	6	150	9.29
Sales Price	No Sales	34.30	-1.6%	10	133	8.67
	Sell at 70%	36.18	+3.8%	9	151	8.91
	Sell at 90%	32.69	-6.2%	7	200	9.57

Finally, the ability to sell surplus inventory is a high-risk, high-reward lever, as the option to sell encourages more proactive procurement by lowering its perceived cost. The results show this strategy is highly sensitive to price. An efficient sales channel (90% recovery price) provides the lowest net cost of any test (-6.2%). However, an inefficient channel (70% recovery price) results in a 3.8% higher net cost, which is significantly worse than having no sales mechanism at all (-1.6% cost). This shows that a sales option is only financially beneficial if the recovery price is high.

V. Discussion

The framework functions essentially as an automated extension of a planner's reasoning. It processes user inputs to generate a probabilistic view of future demand and lead times, quantifying the likelihood of having parts available when needed. This allows the model to use computing power to relieve the planner from manually calculating every possible scenario. It explicitly returns the trade-off between the cost of purchasing safety and the risk of saving money by not buying. When inputs like penalty costs are accurate, this provides a precise optimization. However, if inputs are uncertain, the framework can still be used to find a robust solution by tuning parameters to balance risk under worst-case conditions.

This adaptability allows the model to serve two distinct functions. Beyond its role as a daily operational tool, it acts as a simulation engine for strategic decision-making. As demonstrated in this study, the system can simulate multi-year periods to generate tactical insights. For instance, if holding costs are viewed as a proxy for warehouse investment, a manager can adjust this variable to observe the resulting shift in the total cost breakdown. This capability transforms the model from a simple ordering system into a test bench where different infrastructure effects and high-level policies can be evaluated before implementation.

The speed required for these simulations relies on the model's myopic design. By optimizing decisions only for the current day, the model sacrifices global long-term optimization for computational efficiency. This is a necessary trade-off that becomes a strength in practice. While a real-world application would run once daily with updated information, this design makes it possible to run multi-year simulations in minutes to test different parameters on the spot. The logic remains efficient because it focuses computational resources on the near future, relying on the rolling nature of the horizon to address distant tasks only as they move closer to the decision window.

VI. Conclusion

This research introduced a granular, end-to-end framework for aircraft spare parts management, directly connecting a fleet's static maintenance plan to dynamic, daily procurement actions. The methodology was presented as a modular, two-stage pipeline: a forecasting component that translates raw maintenance tasks into a probabilistic, time-phased demand signal, and a procurement component that acts on this signal using a transparent, cost-based optimization logic.

The primary achievement of this work is the demonstration of a complete, auditable, and task-specific management system. The forecasting methodology provides a repeatable process for generating a demand signal even in data-scarce "cold start" scenarios, using NLP and clustering to create a logical foundation for reliability modeling. The procurement model's core contribution is its transparent, cost-based logic that explicitly mimics a planner's trade-offs. By evaluating the expected costs of buying, waiting, or even selling surplus stock, it provides traceable recommendations. This model proved its financial viability, reducing total net costs by 17.5% and 9.2% compared to fully conservative and periodic-review benchmarks, respectively.

Furthermore, maintenance optimization has shifted toward date-specific and operation-aware planning. This framework brings that same level of detail to spare parts procurement by utilizing all available schedule data. This alignment is more than just an algorithmic improvement; it is a prerequisite for the future adoption of Equalized Maintenance Programs. Because EMPs rely on tight, frequent interventions, the supply chain must operate with the same precision. Without this specific link between tasks and parts, the stability required for an equalized schedule remains out of reach.

Beyond its performance and detail, the framework's value was demonstrated as a decision-support tool for generating actionable insights. The analysis provided quantifiable evidence for several non-linear trade-offs. It confirmed that deliberately accepting a small, controlled number of service failures is financially optimal when the cost of perfect protection is prohibitively high. The model's logic also proved highly robust; when fed biased demand forecasts, its penalty-driven logic automatically compensated by increasing or decreasing regular orders to maintain a safe inventory level. Finally, the analysis identified optimal "sweet spots" for internal policies, such as the perceived penalty multiplier, and quantified the high-risk, high-reward nature of a surplus-selling mechanism. This foundation can be extended through expert validation of the task clusters or by expanding the model to optimize procurement system-wide.

References

Roda, I., Macchi, M., Fumagalli, L., and Viveros, P., "A review of multi-criteria classification of spare parts: From literature analysis to industrial evidences," *Journal of Manufacturing Technology Management*, Vol. 25, No. 4, 2014, pp. 528–549.

Bacchetti, A., and Saccani, N., "Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice," *Omega*, Vol. 40, No. 6, 2012, pp. 722–737.

Ayu Nariswari, N. P., Bamford, D., and Dehe, B., "Testing an AHP model for aircraft spare parts," *Production Planning & Control*, Vol. 30, No. 4, 2019, pp. 329–344.

Kang, Z., Marandi, A., Basten, R. J., and de Kok, T., "Robust spare parts inventory management," *Available at SSRN 4553430*, 2023.

Hekimoğlu, M., Kök, A. G., and Şahin, M., "Stockout risk estimation and expediting for repairable spare parts," *Computers & Operations Research*, Vol. 138, 2022, p. 105562.

Qin, Y., Ma, H.-L., Chan, F. T., and Khan, W. A., "A scenario-based stochastic programming approach for aircraft expendable and rotable spare parts planning in MRO provider," *Industrial Management & Data Systems*, Vol. 120, No. 9, 2020, pp. 1635–1657.

Erkoc, M., and Ertogral, K., "Overhaul planning and exchange scheduling for maintenance services with rotable inventory and limited processing capacity," *Computers & Industrial Engineering*, Vol. 98, 2016, pp. 30–39.

Buyukkaramikli, N. C., van Ooijen, H. P., and Bertrand, J. W. M., "Integrating inventory control and capacity management at a maintenance service provider," *Annals of Operations Research*, Vol. 231, 2015, pp. 185–206.

A. Rock Algorithm

Algorithm 1: ROCK (RObust Clustering using linKs)

```
Input: Set of tasks $T = \{t_1, t_2, ..., t_n\}$; desired clusters $N_{clusters}$; threshold $\theta$; exponent $\epsilon$
Output: Partition of $T$ into $N_{clusters}$
Parameters used in this study:
$N_{clusters} \leftarrow 7$
$\theta \leftarrow 0.05$
$\epsilon \leftarrow 2.46$
Similarity function ← Jaccard coefficient
// Neighbor and Link Calculation
for each pair $(t_i, t_j)$ do
    if $J(t_i, t_j) \geq \theta$ then
        Add $t_j$ to $Neighbors(t_i)$ and $t_i$ to $Neighbors(t_j)$;
for each pair $(t_i, t_j)$ do
    $link(t_i, t_j) \leftarrow |Neighbors(t_i) \cap Neighbors(t_j)|$;
// Hierarchical Clustering using Goodness Measure
Initialize singleton clusters $C_i = \{t_i\}$; set $C = \{C_1, ..., C_n\}$;
Compute $g(C_i, C_j)$ for all pairs where $link[C_i, C_j] > 0$;
while $|C| > N_{clusters}$ do
    Select $(C_i, C_j)$ with highest $g(C_i, C_j)$;
    Merge $C_{new} \leftarrow C_i \cup C_j$;
    Update $C$ and recompute $g$ values involving $C_{new}$;
return $N_{clusters}$ final clusters;
```

B. Notation

The notation below serves as a reference for the decision logic in Figure 4 and the evaluation procedure in Algorithm 2.

Indices and Sets

$m \in M$	Index for a specific maintenance task (and its associated spare part).
d	The current calendar day (decision epoch).
τ	Index for a future day in the planning horizon ($\tau > d$).
ℓ	Index for the past day on which an open order was placed ($\ell < d$).
$j \in J^m$	Index of a specific maintenance occurrence for task m.
$\mathcal{T}$	Short-term evaluation window (exact scenario enumeration).
$\mathcal{T}_{\text{LT}}$	Long-term evaluation window (moment propagation).
$\mathcal{S}$	Set of all combined demand and arrival scenarios over $\mathcal{T}$.

Decision Variables (for task m on day d)

$x^{\text{reg}}_{m,d}$	Quantity of a regular order.
$x^{\text{expr}}_{m,d}$	Quantity of an express order.
$x^{\text{sell}}_{m,d}$	Quantity of surplus inventory sold.

State and Tracking Variables

i^m_d	On-hand inventory level at the *end* of day d.
$A^{\text{expected}}_{m,d}$	Set of all open orders. Tracks the specific probabilistic arrival dates for each order placed on day ℓ.
$\mathcal{E}^m_\tau$	Probability Mass Function (PMF) of spare part demand on future day τ.
$\mathcal{I}^m_\tau$	Forecasted PMF of inventory levels on future day τ.
μ_τ, σ^2_τ	Mean and variance of the forecasted inventory distribution $\mathcal{I}^m_\tau$.
$\alpha^{\text{pre}}_{m,\tau}$	Fixed quantity of pre-procurement arrivals scheduled to arrive on day τ.

C. ROCK Cluster Profiles

Table 4. Profiles of ROCK Clusters (Top 3 Attribute Values)

Cluster (Size)	Source Group	Task Code Group	Zone Grp.	Interval Freq.	Failure Mode	Comp. Mvmt.	Access Lvl.	Outcome Expectation
1 (38)	Group 3: 50.0% Group 1: 50.0%	Group 1: 100.0%	1: 57.9% 2: 42.1%	1 to 2: 36.8% <1 time: 34.2% 3 to 9: 23.7%	Contam./Blockage: 76.3% Mechanical: 10.5% Electrical: 7.9%	Static: 84.2% Moving: 13.2% N/A: 2.6%	Disassembly Req.: 47.4% Limited Internal: 28.9% N/A: 15.8%	Replace Cons.: 71.1% Minor Servicing: 15.8% Overhaul/Bench Test: 7.9%
2 (58)	Group 3: 65.5% Group 1: 34.5%	Group 2: 55.2% Group 3: 24.1% Group 4: 19.0%	1: 53.4% 2: 44.8% 3 or 4: 1.7%	<1 time: 48.3% 1 to 2: 25.9% 10+: 13.8%	Contam./Blockage: 62.1% Leakage: 22.4% Electrical: 12.1%	Static: 87.9% Moving: 8.6% N/A: 3.4%	Limited Internal: 75.9% External Surface: 15.5% Disassembly Req.: 5.2%	Leak/Contam. Cont.: 50.0% Funct. Cont.: 22.4% Minor Servicing: 22.4%
3 (473)	Group 3: 83.3% Group 2: 10.6% Group 1: 6.1%	Group 4: 79.5% Group 5: 20.1% Group 3: 0.2%	2: 58.8% 1: 26.2% 3 or 4: 10.4%	<1 time: 82.2% 1 to 2: 12.1% 3 to 9: 4.4%	Structural: 87.1% Mechanical: 3.4% Electrical: 3.2%	Static: 85.4% Moving: 13.1% Intermittent: 0.8%	Limited Internal: 71.7% External Surface: 22.2% Full Internal: 5.9%	Visual Struct. Cont.: 96.0% Minor Servicing: 2.1% Funct. Cont.: 1.1%
4 (145)	Group 3: 79.3% Group 1: 20.7%	Group 3: 83.4% Group 5: 9.0% Group 2: 6.2%	1: 94.5% 2: 4.8% 3 or 4: 0.7%	<1 time: 35.2% 1 to 2: 31.7% 10+: 24.8%	Electrical: 73.1% Leakage: 11.7% Mechanical: 10.3%	Static: 72.4% N/A: 15.2% Moving: 11.0%	External Surface: 75.2% N/A: 20.0% Limited Internal: 3.4%	Funct. Cont.: 89.7% Visual Struct. Cont.: 3.4% Leak/Contam. Cont.: 3.4%
5 (126)	Group 1: 62.7% Group 3: 36.5% Group 2: 0.8%	Group 3: 77.8% Group 4: 11.9% Group 2: 9.5%	1: 69.8% 3 or 4: 15.9% 2: 11.1%	<1 time: 65.1% 1 to 2: 24.6% 10+: 5.6%	Mechanical: 82.5% Electrical: 10.3% Leakage: 7.1%	Moving: 88.9% Intermittent: 5.6% Static: 4.8%	Limited Internal: 48.4% N/A: 25.4% External Surface: 23.0%	Funct. Cont.: 93.7% Visual Struct. Cont.: 4.0% Overhaul/Bench Test: 1.6%
6 (68)	Group 3: 52.9% Group 1: 47.1%	Group 2: 67.6% Group 4: 13.2% Group 5: 11.8%	2: 64.7% 1: 23.5% 3 or 4: 11.8%	1 to 2: 39.7% 3 to 9: 27.9% <1 time: 23.5%	Mechanical: 73.5% Contam./Blockage: 10.3% N/A: 8.8%	Moving: 89.7% N/A: 5.9% Intermittent: 2.9%	Limited Internal: 66.2% N/A: 17.6% External Surface: 10.3%	Minor Servicing: 50.0% Funct. Cont.: 30.9% Visual Struct. Cont.: 13.2%
7 (206)	Group 3: 59.7% Group 2: 38.3% Group 1: 1.9%	Group 5: 89.8% Group 4: 9.7% Group 3: 0.5%	2: 47.1% 1: 45.6% 3 or 4: 4.4%	<1 time: 46.1% 1 to 2: 33.5% 10+: 11.7%	Structural: 92.7% Leakage: 2.9% Mechanical: 1.5%	Static: 92.2% Moving: 6.8% N/A: 1.0%	External Surface: 91.3% Limited Internal: 8.7%	Visual Struct. Cont.: 96.6% Leak/Contam. Cont.: 1.9% Funct. Cont.: 1.5%
8 (20)	Group 3: 70.0% Group 1: 30.0%	Group 5: 95.0% Group 4: 5.0%	1: 100.0%	<1 time: 95.0% 1 to 2: 5.0%	Contam./Blockage: 40.0% Leakage: 30.0% No Failure: 25.0%	Static: 75.0% Moving: 25.0%	Disassembly Req.: 65.0% Limited Internal: 15.0% N/A: 15.0%	Replace Cons.: 50.0% Visual Struct. Cont.: 20.0% Minor Servicing: 15.0%

Note on Abbreviations: N/A = Not Applicable; Contam. = Contamination; Funct. Cont. = Functional Continue; Struct. = Structural; Comp. Mvmt. = Component Movement; Access Lvl. = Access Level; Disassembly Req. = Disassembly Required; Leak/Contam. Cont. = Leakage or Contamination Continue; Visual Struct. Cont. = Visual Structural Continue; Replace Cons. = Replace Consumable.

D. Expected Cost Evaluation Algorithm

Algorithm 2: Expected Cost Evaluation for Procurement Decision $(x^{\text{reg}}, x^{\text{sell}})$ on Day d

Input: Part m; Current day d; State: i^m_{d-1} (inventory), $A^{\text{expected}}_{m,d-1} = \{(q_\ell, \pi_\ell, \ell) : \ell < d\}$ (open orders with quantity, PMF, order day);
Action: x^{reg} (regular order), x^{sell} (sell quantity); Data: $\{\mathcal{E}^m_\tau\}_{\tau \geq d}$ (demand PMFs), $\{d^m_j\}_{j \in J^m}$ (maint. days),
Π^{LT}_m (lead time PMF), $\{\alpha^{\text{pre}}_{m,\tau}\}_\tau$ (pre-procurement); Costs: P^m, h_m, S^m, PEN^m, C^m_{reg}, C^m_{expr}; Risk tolerance: α

Output: V^* (total expected cost), x^{expr} (express order), $A^{\text{expected}}_{m,d}$ (updated orders), $\{\mathcal{I}^m_\tau\}_{\tau \in \mathcal{T}}$ (inventory PMFs)

```
// State transition from $d-1$ to $d$
$i^m_d \leftarrow i^m_{d-1} - x^{\text{sell}} + \alpha^{\text{pre}}_{m,d} + \sum_{\ell<d} q_\ell \cdot \mathbf{1}\{\text{arrival realized on } d\}$ ;    // Apply action and arrivals
$A^{\text{expected}}_{m,d} \leftarrow \emptyset$ ;    // Rebuild open orders
for $(q_\ell, \pi_\ell, \ell) \in A^{expected}_{m,d-1}$ do
    if $\sum_{\tau>d} \pi_\ell(\tau) > 0$ then
        $\pi'_\ell(\tau) \leftarrow \pi_\ell(\tau) / \sum_{s>d} \pi_\ell(s)$ for all $\tau > d$ ;    // Condition on not arrived yet
        $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \cup \{(q_\ell, \pi'_\ell, \ell)\}$
if $x^{reg} > 0$ then
    $A^{\text{expected}}_{m,d} \leftarrow A^{\text{expected}}_{m,d} \cup \{(x^{\text{reg}}, \Pi^{\text{LT}}_m, d)\}$ ;    // Add new order
// Define evaluation windows
$\mathcal{T} \leftarrow [d+1, \min\{d + \max_\tau\{\tau : \Pi^{\text{LT}}_m(\tau) > 0\}, d_{\text{end}}\}]$ ;    // Short-term: possible arrival window
$\mathcal{T}_{\text{LT}} \leftarrow [\max(\mathcal{T}) + 1, d_{\text{end}}]$ ;    // Long-term: beyond arrival window
// Construct scenario space over $\mathcal{T}$
$\mathcal{A} \leftarrow \{a : A^{\text{expected}}_{m,d} \times \mathcal{T} \to \{0,1\}\}$ where $a_{\ell,\tau} = 1$ iff order $\ell$ arrives on day $\tau$
$\mathcal{A}_{\text{valid}} \leftarrow \{a \in \mathcal{A} : \sum_\tau a_{\ell,\tau} \leq 1 \text{ for each } \ell\}$ ;    // Each order arrives at most once
$\mathcal{D} \leftarrow \prod_{\tau \in \mathcal{T}} \text{support}(\mathcal{E}^m_\tau)$; $\mathcal{S} \leftarrow \mathcal{A}_{\text{valid}} \times \mathcal{D}$ ;    // All demand realizations and complete scenario set
// Compute inventory PMF at each $\tau \in \mathcal{T}$ via scenario aggregation
for $\tau \in \mathcal{T}$ do
    $\mathcal{I}^m_\tau \leftarrow \{\}$ ;    // Initialize as empty PMF
    for $(a, e) \in \mathcal{S}$ do
        $Q^{(a)}_\tau \leftarrow \sum_{(\ell,t) \leq \tau} a_{\ell,t} \cdot q_\ell$ ;    // Total arrivals by $\tau$ in scenario $a$
        $D^{(e)}_\tau \leftarrow \sum_{t \leq \tau} e_t$ ;    // Total demand by $\tau$ in scenario $e$
        $I^{(a,e)}_\tau \leftarrow i^m_d + Q^{(a)}_\tau + \sum_{t \leq \tau} \alpha^{\text{pre}}_{m,t} - D^{(e)}_\tau$ ;    // Inventory level
        $p^{(a,e)} \leftarrow \left[\prod_{\ell,t} \pi_\ell(t)^{a_{\ell,t}} \cdot \left(1 - \sum_{s \leq \tau} \pi_\ell(s)\right)^{1 - \sum_s a_{\ell,s}}\right] \cdot \left[\prod_t p^{\mathcal{E}}_t(e_t)\right]$
        $\mathcal{I}^m_\tau[I^{(a,e)}_\tau] \leftarrow \mathcal{I}^m_\tau[I^{(a,e)}_\tau] + p^{(a,e)}$ ;    // Aggregate probabilities by inventory level
    $\mu_\tau \leftarrow \sum_{(q,p) \in \mathcal{I}^m_\tau} q \cdot p$;    $\sigma^2_\tau \leftarrow \sum_{(q,p) \in \mathcal{I}^m_\tau} (q - \mu_\tau)^2 \cdot p$ ;    // Compute moments
// Express order optimization (activated only if $d = d^m_j - 1$ for some $j$)
$x^{\text{expr}} \leftarrow 0$ ;    // Default: no express order
if $\exists j \in J^m : d = d^m_j - 1$ then
    Define $\Psi(q, e) \equiv \text{PEN}^m \cdot \mathbf{1}\{e > q\} + P^m \cdot \max\{e - q, 0\}$ ;    // Shortfall cost function
    Define $C_{\text{expr}}(x) \equiv x(P^m + C^m_{\text{expr}}) + \sum_{(q,p_q) \in \mathcal{I}^m_{d+1}} \sum_{(e,p_e) \in \mathcal{E}^m_{d+1}} p_q p_e \cdot \Psi(q + x, e)$
    $x^{\text{expr}} \leftarrow \arg\min_{x \in \{0,1,\dots,\bar{x}_{\max}\}} C_{\text{expr}}(x)$ ;    // Minimize expected cost
    if $x^{expr} > 0$ then
        for $\tau > d$ do
            $\mathcal{I}^m_\tau \leftarrow \{(q + x^{\text{expr}}, p) : (q, p) \in \mathcal{I}^m_\tau\}$ ;    // Add express quantity to all inventory levels
// Short-term cost accumulation over $\mathcal{T}$
$V_{\text{hold}} \leftarrow \sum_{\tau \in \mathcal{T}} h_m \cdot \mu_\tau$ ;    // Expected holding cost
$V_{\text{pur}} \leftarrow x^{\text{reg}}(P^m + C^m_{\text{reg}}) + x^{\text{expr}}(P^m + C^m_{\text{expr}})$ ;    // Purchase and shipping
$V_{\text{pen}} \leftarrow \sum_{j : d^m_j \in [d, \max(\mathcal{T})]} \sum_{(q,p_q) \in \mathcal{I}^m_{d^m_j}} \sum_{(e,p_e) \in \mathcal{E}^m_{d^m_j}} p_q p_e \cdot \Psi(q, e)$ ;    // Expected penalties
// Long-term analysis via moment propagation over $\mathcal{T}_{\text{LT}}$
$(\mu_t, \sigma^2_t) \leftarrow (\mu_{\max(\mathcal{T})}, \sigma^2_{\max(\mathcal{T})})$ ;    // Initialize from end of short-term
for $t \in \mathcal{T}_{LT}$ do
    $\mu_t \leftarrow \mu_{t-1} + \alpha^{\text{pre}}_{m,t} - \mathbb{E}[\mathcal{E}^m_t]$;    $\sigma^2_t \leftarrow \sigma^2_{t-1} + \text{Var}[\mathcal{E}^m_t]$
// Projected sale using Cantelli's inequality
$q^* \leftarrow \lfloor \min_{t \in \mathcal{T}_{\text{LT}}} \mu_t \rfloor$;    $t^* \leftarrow \infty$ ;    // Identify stable excess
if $q^* > 0$ and $P^m \geq 500$ then
    $t^* \leftarrow \min\{t \in \mathcal{T}_{\text{LT}} : \frac{\sigma^2_t}{\sigma^2_t + (\mu_t - q^*)^2} \leq \alpha\}$ ;    // First safe potential selling day
$V_{\text{LT}} \leftarrow \sum_{t \in \mathcal{T}_{\text{LT}}}^{t^*-1} h_m \mu_t + \sum_{t^*}^{d_{\text{end}}} h_m \max\{\mu_t - q^*, 0\} - \mathbf{1}\{t^* < \infty\} q^* S^m$
// Total expected cost
$V^* \leftarrow V_{\text{hold}} + V_{\text{pur}} + V_{\text{pen}} + V_{\text{LT}} - x^{\text{sell}} \cdot S^m$
return $V^*$, $x^{expr}$, $A^{expected}_{m,d}$, $\{\mathcal{I}^m_\tau\}_{\tau \in \mathcal{T}}$
```