REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–

ESSA Tlemcen-تلمسان

المدرسة العليا في العلوم التطبيقية
École Supérieure en Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
ـ تلمسان ـ

Final Degree Project

For obtaining the Master diploma

Field : Industrial engineering
Major: Industrial management and logistics

Presented by : Mohammed Abdelkader EL AHMAR

Subject

A Stochastic MILP and Metaheuristic framework for Circular Supply Chain Optimisation Under Demand And Yield Uncertainty

Publicly defended, on 22/06/2025 , in front of the jury composed of:

Mr Mohammed Bennekrouf	Associate Professor	ESSA. Tlemcen	President
Mr Fouad Maliki	Associate Professor	ESSA. Tlemcen	Thesis Director
Mr Taha Arbaoui	Professor	INSA Lyon	Thesis co-Director
Mrs Nour El Houad Sekkal	Assistant Professor	ESSA. Tlemcen	Examiner 1
Mr Haddou Benderbal Hichem	Associate Professor	Aix-Marseille Uni	Examiner 2

Année universitaire : 2025 /2026


بسم الله الرحمن الرحيم

Dedication

This work is dedicated to my loving parents, Mr. Miloud El Ahmar and Mrs. Samia Faraoun, who have stood by me at every stage of my life. I extend my deepest gratitude to them for their endless support and for providing me with every opportunity to pursue my dreams. Their unconditional love has shaped me into the man I am today, and I will remain eternally grateful.

To my little sister, Hidayet El Ahmar, my greatest source of support and hope: Thank you for your lifelong encouragement and for filling my life with cherished moments I will always hold dear.

To my grandparents, Mohammed Faraoun and Fatiha Gerryvile: Thank you for being the foundational pillars of my life and for always providing a guiding hand to lift me up whenever I stumble.

I also wish to honor the memory of my late grandfather, Abdelkader El Ahmar. I wish more than anything that he could be here to see this moment, but I know his love guides me still."

I also extend my heartfelt thanks to my extended family—my aunts, uncles, and cousins—for their love, respect, and constant encouragement. May this humble work reflect my sincere gratitude for their unwavering support.

To my classmates: Thank you for making these past three years the best of my life, filled with joy and wonderful memories.

I would also like to express my sincere appreciation to my professor, Fouad Maliki. You have been a true mentor to me throughout my academic journey, pushing me to achieve milestones I never could have imagined.

Lastly, to everyone who contributed in any way to the completion of this work: your guidance, support, and presence were essential throughout this journey. I thank you from the bottom of my heart for helping to bring this project to life.dédicaces

"We don't get to choose how we start in this life. Real greatness is what you do with the hand you're dealt."

— Victor Sullivan, *Uncharted Series*

Acknowledgement

First and foremost, I express my profound gratitude to Allah, the Almighty, for granting me the strength and perseverance to successfully complete this Master's thesis.

I wish to express my deepest appreciation to my Thesis Director, Mr. Fouad Maliki . His continuous guidance, constant availability, and expertise in Operations Research were invaluable in providing the essential framework for this study.

I am equally grateful to my Co-Supervisor, Professor Taha Arbaoui . His critical insights and extensive knowledge significantly enriched the theoretical and practical quality of this work, helping me navigate its analytical challenges.

It is a profound honor that Mr. Mohammed Bennekrouf has graciously accepted to preside over the jury. I sincerely thank him for his interest in my research and his leadership during the defense.

Finally, my sincere gratitude extends to my examiners, Mrs. Nour El Houad Sekkal and Mr. Haddou Benderbal Hichem . I deeply appreciate the time they dedicated to evaluating this thesis and their constructive feedback, which will greatly benefit my future endeavors in Supply Chain Management.

Abstract

Circular Supply Chains (CSCs) are essential for sustainable resource management, yet their efficiency is frequently undermined by operational unpredictability. This thesis addresses this challenge by extending a deterministic CSC model into a robust, uncertainty-aware framework. The primary objective is to optimize disassembly and remanufacturing operations while explicitly accounting for two major stochastic elements: customer demand fluctuations and disassembly yield variability.

To capture these complexities, a Stochastic Mixed Integer Linear Programming (MILP) model is developed. Probability distributions are used to characterize the uncertain parameters, and a Monte Carlo Simulation approach generates comprehensive stochastic scenarios. Recognizing the computational intractability of solving large-scale stochastic MILP instances with exact methods, the study designs and implements an advanced metaheuristic optimization framework. Extensive computational experiments on synthetically generated datasets are conducted to benchmark the metaheuristic against exact MILP solutions.

Graphical and quantitative analyses reveal that the proposed stochastic framework significantly enhances operational robustness compared to deterministic models, effectively hedging against risks like stockouts and inefficient resource allocation. Furthermore, the metaheuristic demonstrates exceptional computational efficiency, delivering high-quality, near-optimal solutions significantly faster than exact solvers. Ultimately, this research bridges stochastic optimization and metaheuristics, providing supply chain managers with scalable, data-driven tools to design highly resilient and economically viable circular networks under persistent uncertainty.

Résumé

Les chaînes d'approvisionnement circulaires (CAC) sont essentielles à la gestion durable des ressources, mais leur efficacité est fréquemment compromise par l'imprévisibilité opérationnelle. Cette thèse relève ce défi en étendant un modèle déterministe de CAC vers un cadre décisionnel robuste intégrant l'incertitude. L'objectif principal est d'optimiser les opérations de désassemblage et de refabrication tout en tenant explicitement compte de deux éléments stochastiques majeurs : les fluctuations de la demande client et la variabilité du rendement de désassemblage.

Afin de capturer ces complexités, un modèle stochastique de Programmation Linéaire en Nombres Entiers Mixte (PLNEM) est développé. Des distributions de probabilité sont utilisées pour caractériser les paramètres incertains, et une approche par simulation de Monte-Carlo génère des scénarios stochastiques complets. Reconnaissant la complexité de résolution d'instances stochastiques à grande échelle par des méthodes exactes, l'étude conçoit et met en œuvre un cadre d'optimisation métaheuristique avancé. Des expériences numériques approfondies sur des jeux de données générés synthétiquement sont menées pour comparer les performances de la métaheuristique aux solutions exactes.

Les analyses graphiques et quantitatives révèlent que le cadre stochastique proposé améliore considérablement la robustesse opérationnelle par rapport aux modèles déterministes, protégeant efficacement le système contre les risques de rupture de stock et d'allocation inefficace des ressources. De plus, la métaheuristique fait preuve d'une efficacité computationnelle exceptionnelle, fournissant des solutions de haute qualité, quasi-optimales, beaucoup plus rapidement que les solveurs exacts. En définitive, cette recherche fait le lien entre l'optimisation stochastique et les métaheuristiques, offrant aux gestionnaires des outils évolutifs et fondés sur les données pour concevoir des réseaux circulaires hautement résilients et économiquement viables face à une incertitude persistante.

الملخص

تُعد سلاسل التوريد الدائرية أساسية لإدارة الموارد المستدامة، إلا أن كفاءتها غالباً ما تتقوض بسبب عدم اليقين التشغيلي. تعالج هذه الأطروحة هذا التحدي من خلال توسيع نموذج حتمي لسلسلة التوريد الدائرية ليصبح إطار عمل قوي يراعي عدم اليقين في اتخاذ القرار. يتمثل الهدف الرئيسي في تحسين عمليات التفكيك وإعادة التصنيع مع مراعاة عنصرين عشوائيين رئيسيين بشكل صريح: تقلبات طلب العملاء، وتغير عائد التفكيك.

ولاستيعاب هذه التعقيدات، تم تطوير نموذج عشوائي للبرمجة الخطية الصحيحة المختلطة (صتخهستخ پيبع). تُستخدم التوزيعات الاحتمالية لتوصيف المعلمات غير المؤكدة، كما يُولد نهج محاكاة رولين هوزيزن سيناريوهات عشوائية شاملة. ونظراً للصعوبة الحسابية في حل المسائل العشوائية واسعة النطاق لنموذج (پيبع) باستخدام الطرق الدقيقة، تصمم الدراسة وتنفذ إطار عمل متقدم للتحسين باستخدام الخوارزميات الفوقية (پتهـرستخس). تم إجراء تجارب حسابية مكثفة باستخدام مجموعات بيانات مُولدة اصطناعياً لمقارنة أداء الخوارزمية الفوقية مع الحلول الدقيقة.

تكشف التحليلات البيانية والكمية أن الإطار العشوائي المقترح يعزز بشكل كبير من المتانة التشغيلية مقارنة بالنماذج الحتمية، مما يقي بفعالية من المخاطر مثل نفاد المخزون وعدم كفاءة تخصيص الموارد. علاوة على ذلك، تُظهر الخوارزمية الفوقية كفاءة حسابية استثنائية، حيث تقدم حلولاً عالية الجودة وشبه مثالية في وقت أسرع بكثير من أدوات الحل الدقيقة. في النهاية، يربط هذا البحث بين التحسين العشوائي والخوارزميات الفوقية، مما يزود مديري سلاسل التوريد بأدوات قابلة للتطوير وقائمة على البيانات لتصميم شبكات دائرية عالية المرونة وذات جدوى اقتصادية في ظل عدم اليقين المستمرا

Contents

List of Figures **14**

List of Tables **15**

Abreviations list **15**

Introduction **17**

1 Optimization Theory **21**
1.1 Introduction to Optimization 21

2 Simulation and Stochastic Methods **23**
2.1 Introduction to Simulation 23
2.2 Types of Simulation 24
2.2.1 Deterministic vs. Stochastic Simulation 24
2.2.2 Continuous vs. Discrete-Event Simulation 24
2.2.3 Agent-Based Simulation 24
2.2.4 System Dynamics 25
2.3 Probability Theory Fundamentals 25
2.3.1 Random Variables and Probability Distributions 25
2.3.2 Expected Value, Variance, and Covariance 25
2.4 Common Probability Distributions 26
2.5 Random Number Generation 26
2.5.1 Pseudo-Random Number Generators 26
2.5.2 Inverse Transform Method 27
2.5.3 Acceptance–Rejection Method 27
2.6 Monte Carlo Simulation 27

2.6.1 Mathematical Foundations . . . 27
2.6.2 Variance Reduction Techniques . . . 28
2.7 Stochastic Processes . . . 29
2.7.1 Markov Chains . . . 29
2.7.2 Poisson Processes . . . 29
2.7.3 Renewal Processes . . . 29
2.7.4 Birth–Death Processes . . . 30
2.8 Stochastic Optimization . . . 30
2.8.1 Chance-Constrained Programming . . . 30
2.8.2 Robust Optimization . . . 30
2.8.3 Two-Stage Stochastic Programming . . . 31
2.8.4 Multi-Stage Stochastic Programming . . . 31
2.9 Discrete-Event Simulation . . . 31
2.9.1 Key Components . . . 32
2.9.2 The Event-Scheduling Algorithm . . . 32
2.10 Applications in Supply Chains and Circular Systems . . . 33
2.10.1 Demand and Lead-Time Uncertainty . . . 33
2.10.2 The Newsvendor Model . . . 33
2.10.3 Circular Supply Chains and Remanufacturing . . . 33
2.11 Comparison of Optimization and Simulation . . . 34
2.11.1 Simulation-Optimization Frameworks . . . 34

3 Stochastic Modeling **36**
3.1 Executive Summary . . . 36
3.2 Model Description . . . 37
3.2.1 Supply Chain Network . . . 37
3.2.2 Sets, Parameters and Decision Variables . . . 37
3.2.3 Stochastic Processes . . . 38
3.3 Mathematical Formulation . . . 38
3.3.1 Non-Anticipative Information Structure . . . 38
3.3.2 Rolling-Horizon MILP Objective . . . 38
3.3.3 Inventory Balance Constraints . . . 38
3.3.4 Capacity and Feasibility Constraints . . . 39
3.4 Sources of Uncertainty . . . 39
3.4.1 Demand Uncertainty . . . 39
3.4.2 Yield Uncertainty . . . 40
3.4.3 Relationship between Uncertain Parameters and the Model 40
3.5 Probability Distribution Selection . . . 41
3.5.1 Demand: Log-Normal Distribution . . . 41

3.5.2 Yield: Beta Distribution . . . 41
3.5.3 Scenario Reduction and Equiprobable Assignment . . . 42
3.5.4 Modelling Assumptions . . . 43
3.6 Computational Setup . . . 43
3.7 Numerical Results . . . 44
3.7.1 Aggregate Performance Metrics . . . 44
3.7.2 Scenario-Level Results . . . 44
3.7.3 Expected Results by Period . . . 46
3.8 Figures and Visual Analysis . . . 47
3.8.1 Stochastic Data Generation . . . 47
3.8.2 Profit Analysis . . . 49
3.8.3 Remanufacturing Activity . . . 51
3.8.4 Dual-Pool Inventory Architecture . . . 53
3.8.5 Operations and Supply Dynamics . . . 53
3.8.6 Information Asymmetry and Yield Effects . . . 55
3.8.7 Full KPI Dashboard . . . 57
3.9 Discussion . . . 57
3.9.1 Value of the Stochastic Approach . . . 57
3.9.2 Remanufacturing Economics . . . 58
3.9.3 Scenario Variability . . . 58
3.9.4 Limitations and Future Work . . . 58
3.10 Conclusion . . . 59

4 Metaheuristics **60**
4.1 Problem Statement and Objective . . . 61
4.1.1 Full Set of Decision Variables . . . 61
4.2 Chromosome Design . . . 61
4.2.1 Encoding Choice: Why Real-Valued? . . . 61
4.2.2 Chromosome Structure . . . 62
4.2.3 Why Exactly These Three Variables? . . . 62
4.3 Greedy Repair Decoder . . . 64
4.3.1 What is a Repair Decoder? . . . 64
4.3.2 Gene-to-Decision Mapping . . . 64
4.3.3 Priority Order Within Each Period . . . 64
4.4 Population Initialisation . . . 65
4.5 Selection Operator . . . 66
4.5.1 Binary Tournament Selection ($k = 3$) . . . 66
4.5.2 Why Tournament, and Why $k = 3$? . . . 66
4.6 Crossover Operator — SBX . . . 67

4.6.1 Simulated Binary Crossover (SBX) . . . 67
4.6.2 The Distribution Index $\eta_c = 15$ . . . 67
4.6.3 Crossover Probability $p_c = 0.90$ . . . 68
4.7 Mutation Operator — Polynomial Mutation . . . 68
4.7.1 Polynomial Mutation Mechanics . . . 68
4.7.2 Per-Gene Mutation Probability: $p_m = 1/n_{\text{genes}}$ . . . 69
4.7.3 Adaptive Mutation Rate . . . 69
4.7.4 Distribution Index $\eta_m = 20$ . . . 70
4.8 Elitism . . . 70
4.9 Diversity Restart Mechanism . . . 70
4.9.1 Why Restarts Are Necessary . . . 70
4.9.2 Trigger Conditions . . . 71
4.9.3 Restart Procedure . . . 71
4.10 Complete Hyperparameter Table . . . 72
4.11 Model Results . . . 72
4.11.1 GA convergence milestones . . . 72
4.11.2 Financial Results Comparison . . . 73
4.12 GA vs MILP — Structural Differences . . . 73
4.12.1 Why the GA Can Match or Exceed the MILP Objective . . . 73
4.12.2 What the GA Cannot Provide . . . 74
4.13 Computational Complexity and Scalability . . . 75
4.13.1 Decode Complexity . . . 75
4.13.2 Operating Strategy Differences . . . 75
4.14 Conclusions . . . 76

List of Figures

3.1 Forward demand fan $D^{f}_{it,s}$ for Battery Modules (N4) and AESC Pouch Cells (N7) across $S = 20$ log-normal scenarios. Each colour represents one scenario. The dashed black line is the deterministic expectation $\bar{D}_t$ used as lookahead for periods $\tau > 0$. 47

3.2 Time-varying yield fan $\alpha_{ij,t,s}$ for disassembly arcs (N0,N4) and (N4,N7). Each period's yield is drawn independently from a Beta(12, 12) distribution scaled by the bill-of-materials quantity β_{ij}. The dashed line marks the expected yield $\bar{\alpha}_{ij}$. 48

3.3 Marginal probability distributions. **Left**: Log-Normal demand for N4 at $t = 1$ with CV=20%. **Right**: Beta yield distribution for arc (N0,N4) aggregated over all (s, t) realisations; both theoretical density and observed histogram are shown. 48

3.4 Cumulative profit trajectories for all $S = 20$ scenarios. The bold blue curve is the expected rolling-horizon profit E[RH]; the red dashed line is the deterministic benchmark. The shaded region spans the min–max scenario range, illustrating profit variability. . . 49

3.5 Terminal cumulative profit for each scenario (bars) versus the wait-and-see upper bound (transparent purple bars). Green bars exceed E[RH]; orange bars fall below. The horizontal lines mark E[RH], DET, and E[WS]. 49

3.6 Value-chain waterfall: DET → E[RH] → EVPI gap → E[WS]. The EVPI of \$9,094 stacked on E[RH] reaches the wait-and-see upper bound. 50

3.7 Period profit distribution across scenarios shown as percentile bands: min–max range (lightest), interquartile range (medium), and expected value (solid line). Negative periods correspond to high disassembly cost without sufficient immediate revenue (inventory build-up phase). 50
3.8 Remanufacturing starts $r_{k,t,s}$ for Battery Modules (N4) and AESC Pouch Cells (N7). All 20 scenarios show positive $r_{k,t}$ Expected values (bold lines) are well below the capacity caps. 51
3.9 Certified-refurb pool inventories $I^{\text{rmfg}}_{k,t,s}$, fed exclusively by previous-period remanufacturing starts $r_{k,t-1}$. The pool is kept separate from the forward inventory 51
3.10 Remanufacturing cost per period across all scenarios. The expected curve (bold red) confirms a non-zero remanufacturing cost every period, with total $E[\text{RMFG Cost}] = \$13,815$. 51
3.11 Expected remanufacturing revenue (green bars), remanufacturing cost (red bars, shown negative), and net remanufacturing margin (dark line) per period. Both are non-zero in every period, verifying economic viability. 52
3.12 Total remanufacturing starts summed over all $T = 50$ periods, for each scenario. Orange: Battery Modules (N4); green: AESC Pouch Cells (N7). Expected totals $E[\Sigma r_{N4}] = 679$, $E[\Sigma r_{N7}] = 1539$ are marked by dashed lines. 52
3.13 Dual-pool inventory dynamics for two representative scenarios. **Top**: forward pool I^{fwd}_{N4} (orange) and certified-refurb pool I^{rmfg}_{N4} (purple). **Bottom**: remanufacturing starts r_{N4} (bars) feeding the refurb pool, with refurb sales Y^{r}_{N4} (green line). 53
3.14 Pack disassembly decisions $q_{N0,t}$ across scenarios (faint lines), expected value (dark bold), and supply curve S_t (blue dotted). The model adapts disassembly to available supply and revealed downstream demand. 53
3.15 **Top**: Expected forward revenue (blue), remanufacturing revenue (stacked green), and total cost (red line) per period. **Bottom**: Expected period profit $E[Z_t]$; green = positive, red = negative. 54
3.16 Stacked expected cost decomposition per period. Four components are visible: disassembly (blue), remanufacturing (green — non-zero ✓), holding (amber), and disposal (grey). Remanufacturing cost grows as the refurb pipeline matures. 54

3.17 Non-anticipativity illustration at $t = 10$: the yield α for the current period is known (green bars), while the yield five periods ahead is not yet revealed (grey bars). The MILP uses $E[\alpha]$ (dashed) for all future lookahead periods. 55
3.18 Effect of yield revelation on forward inventory and period profit for two scenarios. **Top**: revealed yield $\alpha_{N0N4,t,s}$ (bars) versus forward inventory I^f_{N4} (blue line). **Bottom**: resulting period profit — low-yield periods reduce inventory and compress revenue in subsequent periods. 55
3.19 Scatter plot of mean yield $\bar{\alpha}_{N0N4}$ (averaged over all $T = 50$ periods) versus terminal cumulative profit for each scenario. The positive trend confirms that higher realised yield directly translates into greater module availability and higher profit. Trend line shown dashed. 56
3.20 Comprehensive KPI dashboard summarising: cumulative profit fan (top-left), revenue decomposition (top-right), remanufacturing starts and pool (mid-left), scenario profit vs WS (mid-centre), value chain waterfall (mid-right), yield and demand uncertainty bands (bottom-left/centre), and expected period profit (bottom-right). . . . 57

4.1 SBX spread factor distribution for various η_c. $\beta < 1$: offspring between parents (exploitation). $\beta > 1$: offspring beyond parents (exploration). $\eta_c = 15$ gives moderate spread — the canonical NSGA-II value. 67
4.2 Polynomial mutation perturbation distribution for various η_m. $\eta_m = 20$ produces mostly small perturbations (fine-tuning) with exponentially rare large jumps (escape from local optima). 69
4.3 Diversity restart logic. Triggered only when both stagnation AND collapsed diversity are detected simultaneously. 71

List of Tables

2.1 Summary of common probability distributions used in simulation. 26
2.2 Comparison of optimization and simulation approaches. 34

3.1 Impact of uncertain parameters on the deterministic model 40
3.2 Summary of Key Performance Metrics 44
3.3 Per-Scenario Summary: Remanufacturing Activity and Profit . . . 44
3.4 Expected Decisions and Profits per Period ($t = 1, \ldots, 50$) 46

4.1 All decision variables in the optimisation problem 61
4.2 Profitability analysis — why excluded variables are greedy-safe . . 63
4.3 Population seed strategies (individuals 0–6) 66
4.4 Selection method comparison 66
4.5 Effect of η_c on offspring behaviour 68
4.6 Mutation probability and expected mutations per individual 69
4.7 All GA hyperparameters, values, and justifications 72
4.8 GA convergence milestones 72
4.9 Full KPI comparison — MILP vs GA ($T = 50$ horizon) 73
4.10 Strategy comparison: MILP just-in-time vs GA build-ahead 74
4.11 Empirical runtime for 125,000 evaluations 75
4.12 Implied operating strategy for each solution 75

Abreviations list

1. MILP : Mixed:Integer Linear Programme
2. EOL : End:of Life
3. BOM : Bill of Materials
4. CBC : Coin:or Branch and Cut (solver)
5. PuLP : Python LP (Linear Programming library)
6. BMS : Battery Management System
7. LBC : Lithium Battery Controller (Pack BMS variant)
8. EV : Electric Vehicle
9. SoH : State of Health
10. AESC : Automotive Energy Supply Corporation
11. LMO : Lithium Manganese Oxide
12. MCMB : Mesocarbon Microbeads
13. PE/PP : Polyethylene/Polypropylene
14. OR : Operations Research
15. LP : Linear Programme
16. MIP/MILP : Mixed-Integer (Linear) Programme
17. VSS : Value of the Stochastic Solution
18. EVPI : Expected Value of Perfect Information
19. EV : Expected Value (solution)

20. WS : Wait-and-See
21. RH : Rolling Horizon
22. SP : Stochastic Programme
23. SAA : Sample Average Approximation
24. MC : Monte Carlo
25. CV : Coefficient of Variation
26. MPC : Model Predictive Control
27. KPI : Key Performance Indicator
28. EOQ : Economic Order Quantity
29. VRP : Vehicle Routing Problem
30. ATEX : Atmosphères Explosibles (hazardous area classification)
31. B2B : Business to Business
32. BNEF : BloombergNEF
33. OEM : Original Equipment Manufacturer
34. ESSA : École Supérieure des Sciences Appliquées
35. INSA : Institut National des Sciences Appliquées
36. DISP : Décision et Information pour les Systèmes de Production (lab)
37. CITI : Centre of Innovation in Telecommunications and Integration of services (lab)
38. CVaR : Conditional Value-at-Risk
39. IQR : Interquartile Range
40. CSCM : Circular Supply Chain Management
41. CLSC : Closed-Loop Supply Chains

Introduction

The integration of circular economy principles into supply chain operations has led to the formalization of Circular Supply Chain Management . Unlike traditional forward supply chains, CSCM encompasses both forward and reverse logistics, creating Closed- Loop Supply Chains that facilitate the continuous flow of materials, information, and capital . Reverse logistics is a critical subset of CSCM, encompassing the collection, inspection, sorting, and processing of returned EOL products. The transition to closed-loop and circular supply chains offers profound economic and environmental benefits. Economically, organizations can reduce procurement costs for raw materials, tap into secondary markets for remanufactured goods, and improve long-term supply chain resilience. Environmentally, CSCM drastically reduces greenhouse gas emis- sions, limits landfill waste, and lowers overall energy consumption compared to traditional manufacturing processes. However, managing these bidirectional flows introduces significant logistical and operational complexities that require sophisticated decision-making tools.

Real-world circular supply chains are characterized by deep uncertainties that significantly complicate strategic and tactical planning. The successful optimization of disassembly and remanufacturing operations is highly contingent upon accommodating these stochastic elements. Two of the most critical sources of uncertainty in CSCM are demand variability and yield fluctuation. Firstly, customer demand for remanufactured products is notoriously volatile. Un- like new products, the market acceptance of remanufactured goods is heavily influenced by consumer perception, fluctuating prices, and the potential cannibalization of primary markets. This demand uncertainty complicates production planning and inventory man- agement, often leading to stockouts or excessive holding costs. Secondly, disassembly and remanufacturing operations are subject to severe yield un- certainty. The quality, condition, and age of returned EOL cores vary drasti-

cally. Con- sequently, the yield rates obtained from disassembly operations—the proportion of com- ponents that are successfully extracted and deemed suitable for remanufacturing—are inherently unpredictable. Operational risks, including varying disassembly times, poten- tial damage during dismantling, and the discovery of unexpected component degradation, further exacerbate yield uncertainty. Failing to account for these stochastic parameters exposes the supply chain to significant operational and financial risks.

To overcome the limitations of deterministic modeling, operations research provides the framework of stochastic optimization. Fundamental concepts in stochastic programming involve integrating probability distributions into mathematical models to represent uncer- tain parameters. By explicitly accounting for uncertainty, decision-makers can formulate strategies that remain robust across a wide range of possible future realizations. Scenario-based modeling is a prevalent approach within stochastic optimization, par- ticularly the two-stage stochastic programming framework [?]. In this thesis, Monte Carlo simulation is employed as a powerful tool for scenario generation. By sampling from the probability distributions governing demand and yield, Monte Carlo methods generate a discrete set of scenarios that approximate the continuous stochastic space. The multi-scenario evaluation allows for the optimization of expected system performance while facilitating risk-aware decision-making. The primary advantage of this stochastic ap- proach over deterministic models is its ability to produce highly resilient supply chain configurations that proactively hedge against yield losses and demand shocks.

While exact stochastic MILP models provide mathematically guaranteed optimal solu- tions, they suffer from severe computational limitations. As the number of scenarios generated by the Monte Carlo simulation increases to adequately capture the underlying uncertainty, the size of the stochastic MILP grows exponentially. This phenomenon, often referred to as the curse of dimensionality, renders exact optimization solvers computation- ally prohibitive for large-scale, real-world instances. To address these scalability issues, heuristic and metaheuristic approaches are heav- ily motivated within the operations research literature. Metaheuristics are higher-level algorithmic frameworks designed to efficiently explore complex, NP-hard solution spaces. By leveraging intelligent search mechanisms inspired by biological or physical phenom- ena, metaheuristics can navigate vast search spaces to identify high-quality, near-optimal solutions within a fraction of the computational time required by exact solvers. Within the context of this thesis, the metaheuristic serves a crucial role in ensuring the practi- cal applicability of the proposed stochastic framework, balancing the trade-off between

computational efficiency and solution quality.

The foundational deterministic MILP model developed for optimizing disassembly and remanufacturing operations provides a rigorous structural basis for supply chain plan- ning. However, its core limitation lies in the assumption of perfect information. When subjected to the stochastic realities of fluctuating market demand and unpredictable dis- assembly yields, the deterministic optimal solutions often become suboptimal, infeasible, or financially perilous. Therefore, there is a pressing need for uncertainty-aware decision support systems that can optimize circular supply chain operations while systematically managing the risks associated with volatile demand and uncertain yields. Furthermore, the computational burden of scaling such stochastic models necessitates the integration of advanced, scalable solution algorithm

CHAPTER 1

Optimization Theory

This chapter provides a comprehensive introduction to the theory and practice of mathematical optimization as it applies to engineering, management science, and operations research. Starting from first principles, the discussion progressively covers linear programming, integer programming, nonlinear and multi-objective optimization, metaheuristic algorithms, decomposition methods, and supply-chain optimization. The mathematical formulations, algorithmic procedures, and theoretical guarantees presented here constitute the analytical foundation upon which the methodological contributions of this thesis rest.

1.1 Introduction to Optimization

Optimization is the mathematical discipline concerned with selecting the "best" element from a set of available alternatives with respect to a specified criterion . In its most general form, optimization requires determining the values of *decision variables* that either minimize or maximize a scalar *objective function*, subject to a system of *constraints* that define the universe of feasible choices. This abstraction is powerful enough to capture an enormous variety of practical problems in engineering, economics, logistics, public policy, and the natural sciences .

The historical origins of optimization trace back to the classical calculus of variations developed in the seventeenth and eighteenth centuries by Leibniz, Euler, and Lagrange, who sought to characterize functions that extremize integral functionals. The modern era of mathematical programming, however, was inaugurated by George B. Dantzig's invention of the simplex algorithm in 1947 and by Leonid Kantorovich's independent formulation of resource-allocation problems in the Soviet Union . Kantorovich's foundational contributions were recognized with the Nobel Memorial Prize in Economic Sciences in 1975. Since then, the field has expanded dramatically, encompassing integer programming, nonlinear program-

ming, stochastic programming, multi-objective optimization, and combinatorial optimization, among many other branches .

The importance of optimization in contemporary engineering and decision-making is immense. In supply chain management, optimization models determine optimal inventory policies, production schedules, and vehicle routes, directly impacting cost and service-level performance. In energy systems, unit commitment and economic dispatch problems are solved as large-scale mixed integer programs. In finance, portfolio optimization balances expected return against risk. In machine learning, loss minimization over millions of parameters is the engine driving deep neural networks. The proliferation of high-performance computing and advanced solver technology has made large-scale optimization tractable for industrial problems that would have been computationally intractable only a few decades ago.

Definition 1.1 (General Optimization Problem)**.** A general optimization problem is defined as:

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}) \tag{1.1}$$

$$\text{subject to} \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m, \tag{1.2}$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p, \tag{1.3}$$

where $\mathcal{X} \subseteq \mathbb{R}n$ is the *ground set*, $f : \mathbb{R}n \to$

Definition 1.2 (Optimal Solution)**.** Solve the LP *relaxation* (ignore integrality) to obtain a lower bound (for minimization).

If the LP solution is integer-feasible, it is optimal for the IP.

Otherwise, select a fractional variable x_j and create two sub-problems (*branches*) by adding $x_j \leq \lfloor x_j^* \rfloor$ and $x_j \geq \lceil x_j^* \rceil$, respectively.

Prune branches whose lower bound exceeds the best known integer solution (*incumbent*).

[t] [1] LP relaxation of IP; $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, integrality constraints Optimal integer solution $\mathbf{x}^*$ and optimal value z^* $z^* \leftarrow +\infty$; $\mathbf{x}^* \leftarrow \emptyset$; initialize priority queue $\mathcal{Q}$ with root node $\mathcal{Q} \neq \emptyset$ Remove node P from $\mathcal{Q}$; solve LP relaxation P P is infeasible **or** LP value $\geq z^*$ **prune** P infeasibility or bound pruning LP solution $\mathbf{x}$ is integer-feasible $z^* \leftarrow \mathbf{c}^\top \mathbf{x}$; $\mathbf{x}^* \leftarrow \mathbf{x}$ update incumbent Choose fractional variable x_j; let $v = x_j$ Add child P_1: constraints of P plus $x_j \leq \lfloor v \rfloor$ Add child P_2: constraints of P plus $x_j \geq \lceil v \rceil$ Push P_1, P_2 onto $\mathcal{Q}$

CHAPTER 2

Simulation and Stochastic Methods

This chapter provides a rigorous treatment of simulation and stochastic modelling as they are applied in operations research and supply chain management. Beginning with foundational concepts of probability theory, the discussion develops the mathematical machinery required for Monte Carlo methods, stochastic processes, and stochastic optimization. The chapter concludes with a systematic comparison of optimization and simulation, and an overview of hybrid simulation-optimization frameworks.

2.1 Introduction to Simulation

Simulation is the process of designing a logical or mathematical model of a real system and conducting experiments with that model to understand its behavior and to evaluate various strategies for its operation . Unlike analytical methods that seek closed-form solutions, simulation allows the analyst to represent virtually arbitrary levels of complexity and randomness, making it an indispensable complement to optimization in operations research .

The modern history of simulation traces its origins to the Second World War, when Stanislaw Ulam and John von Neumann developed Monte Carlo methods for neutron diffusion calculations at Los Alamos National Laboratory . The first general-purpose simulation language, GPSS (General Purpose Simulation System), was released by IBM in 1961, followed by SIMSCRIPT, SLAM, and Arena in subsequent decades . Today, simulation tools range from specialized discrete-event engines (Arena, AnyLogic, SimPy) to general programming frameworks that incorporate statistical sampling, parallel computing, and machine learning .

The importance of simulation in operations research stems from three key capabilities: (1) it can model systems too complex for analytical treatment; (2) it enables

risk-free experimentation on a model rather than on the real system; and (3) it provides detailed performance statistics—means, variances, quantiles, confidence intervals—that support well-calibrated managerial decisions under uncertainty . In supply chains, simulation has been applied to evaluate inventory policies under stochastic demand, assess the resilience of distribution networks, and guide capacity planning decisions .

2.2 Types of Simulation

Simulation models are classified along several dimensions: the nature of the state variables, the treatment of time, and the role of randomness .

2.2.1 Deterministic vs. Stochastic Simulation

In *deterministic simulation*, no random variables are involved; the system evolves in a fully predictable manner given initial conditions. Examples include computational fluid dynamics and numerical integration of ordinary differential equations. In *stochastic simulation*, at least one model input or transition is governed by a random variable, so each replication produces a different outcome .

2.2.2 Continuous vs. Discrete-Event Simulation

Continuous simulation tracks state variables that change continuously over time, governed by differential equations—e.g., epidemic spread or reservoir water levels. *Discrete-event simulation* (DES) advances the simulation clock from event to event, changing state only at the instants of those events . DES is especially well-suited to queuing systems, manufacturing lines, and supply chains, where the state changes only when a customer arrives, a machine fails, or an order is dispatched.

2.2.3 Agent-Based Simulation

Agent-based simulation (ABS) models a system as a collection of autonomous agents, each following local rules and interacting with other agents and their environment . Emergent global behaviors—such as traffic congestion or epidemic waves—arise from micro-level interactions, making ABS particularly powerful for complex adaptive systems.

2.2.4 System Dynamics

System dynamics models a system as a network of stocks (accumulations) and flows (rates of change) interconnected by feedback loops . The bullwhip effect in supply chains, where small demand variability amplifies upstream in the form of large order fluctuations, is a classical system-dynamics phenomenon extensively studied using this paradigm .

2.3 Probability Theory Fundamentals

Stochastic simulation and optimization rest on a rigorous foundation of probability theory. The core objects are introduced below.

2.3.1 Random Variables and Probability Distributions

Definition 2.1 (Random Variable 65)**.** A *random variable* X defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a measurable function $X : \Omega \to \mathbb{R}$. It is *discrete* if its range is countable, and *continuous* if its distribution is characterized by a density function.

2.3.2 Expected Value, Variance, and Covariance

Definition 2.2 (Expected Value 65)**.** The *expected value* (mean) of a random variable X is:

$$\mathbb{E}[X] = \begin{cases} \displaystyle\sum_{x} x\, p(x) & \text{(discrete)}, \\ \displaystyle\int_{-\infty}^{\infty} x\, f(x)\, \mathrm{d}x & \text{(continuous)}. \end{cases} \tag{2.1}$$

Definition 2.3 (Variance and Covariance 65)**.** The *variance* of X is

$$\operatorname{Var}(X) = \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2. \tag{2.2}$$

The *covariance* of X and Y is

$$\operatorname{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \tag{2.3}$$

The *correlation coefficient* is $\rho(X, Y) = \operatorname{Cov}(X, Y) / \sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}$.

Two random variables are *independent* if $\mathbb{P}(X \le x, Y \le y) = F_X(x) F_Y(y)$ for all x, y; independence implies $\operatorname{Cov}(X, Y) = 0$, but the converse is not true in general .

2.4 Common Probability Distributions

Table 2.1 summarizes the most important probability distributions used in simulation and stochastic modeling .

Table 2.1: Summary of common probability distributions used in simulation.

Distribution	**Support**	**PDF / PMF** $f(x)$	**Mean**	**Variance**	**CDF** $F(x)$
Uniform(a,b)	$[a,b]$	$\dfrac{1}{b-a}$	$\dfrac{a+b}{2}$	$\dfrac{(b-a)^2}{12}$	$\dfrac{x-a}{b-a}$
Bernoulli(p)	$\{0,1\}$	$p^x(1-p)^{1-x}$	p	$p(1-p)$	—
Binomial(n,p)	$\{0,\dots,n\}$	$\binom{n}{x}p^x(1-p)^{n-x}$	np	$np(1-p)$	(incomplete beta)
Poisson(λ)	$\{0,1,2,\dots\}$	$\dfrac{e^{-\lambda}\lambda^x}{x!}$	λ	λ	(incomplete gamma)
Exponential(λ)	$[0,\infty)$	$\lambda e^{-\lambda x}$	$\dfrac{1}{\lambda}$	$\dfrac{1}{\lambda^2}$	$1-e^{-\lambda x}$
Normal(μ,σ^2)	$(-\infty,\infty)$	$\dfrac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ^2	$\Phi\!\left(\frac{x-\mu}{\sigma}\right)$
Weibull(k,λ)	$[0,\infty)$	$\dfrac{k}{\lambda}\left(\dfrac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}$	$\lambda\,\Gamma\!\left(1+\frac{1}{k}\right)$	$\lambda^2\!\left[\Gamma\!\left(1+\frac{2}{k}\right)-\Gamma\!\left(1+\frac{1}{k}\right)^2\right]$	$1-e^{-(x/\lambda)^k}$

Remark 2.1. The *exponential distribution* with rate λ is the unique continuous distribution with the *memoryless property*: $\mathbb{P}(X > s+t \mid X > s) = \mathbb{P}(X > t)$, making it the natural model for inter-arrival times in Poisson processes . The *Weibull distribution* generalizes the exponential and is widely used to model component lifetimes and failure-time data in reliability engineering .

2.5 Random Number Generation

Simulation requires large quantities of pseudo-random numbers that mimic independent samples from the Uniform$(0,1)$ distribution . These are subsequently transformed into samples from any target distribution.

2.5.1 Pseudo-Random Number Generators

The most widely used class of pseudo-random number generators (PRNGs) is the *Linear Congruential Generator* (LCG), defined by the recurrence :

$$X_{n+1} = (a\,X_n + c) \bmod m, \qquad U_n = X_n/m, \tag{2.4}$$

where m (modulus), a (multiplier), c (increment), and X_0 (seed) are non-negative integers. The Hull–Dobell theorem provides necessary and sufficient conditions for $\{U_n\}$ to have a full period of m: (i) $\gcd(c,m) = 1$; (ii) $a \equiv 1 \pmod p$ for every prime p dividing m; (iii) if $4 \mid m$ then $4 \mid (a-1)$. Modern simulation software

typically employs more sophisticated generators such as the Mersenne Twister, which achieves a period of $2^{19937} - 1$.

2.5.2 Inverse Transform Method

The *inverse transform method* converts a uniform random variate $U \sim \text{Uniform}(0, 1)$ into a sample from any distribution with CDF F via

$$X = F^{-1}(U). \tag{2.5}$$

This follows immediately from the probability integral transform: if F is continuous and strictly increasing, $F(X) \sim \text{Uniform}(0, 1)$. For the exponential distribution, $F^{-1}(u) = -\ln(1 - u)/\lambda$, providing an exact and efficient sampler.

2.5.3 Acceptance–Rejection Method

When F^{-1} is not available in closed form, the *acceptance–rejection* method generates samples from a target density f using a *proposal density* g such that $f(x) \leq c\, g(x)$ for all x and some constant $c \geq 1$:

1. Generate $Y \sim g$ and $U \sim \text{Uniform}(0, 1)$.
2. If $U \leq f(Y)/[c\, g(Y)]$, accept $X = Y$; otherwise, reject and repeat.

The expected number of iterations per accepted sample is c. Choosing g close to f minimizes c and maximizes efficiency.

2.6 Monte Carlo Simulation

Monte Carlo (MC) simulation is a computational technique that estimates quantities of interest by averaging the results of a large number of random experiments . It is one of the most versatile tools in computational science, with applications spanning integration, optimization, finance, and queuing theory.

2.6.1 Mathematical Foundations

Let $\theta = \mathbb{E}[h(X)]$ for some function h and random variable $X \sim F$. The Monte Carlo estimator based on N independent and identically distributed (i.i.d.) samples $X_1, \ldots, X_N \sim F$ is:

$$\hat{\theta}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i). \tag{2.6}$$

By the *Strong Law of Large Numbers*, $\hat{\theta}_N \to \theta$ almost surely as $N \to \infty$. The convergence rate is quantified by the Central Limit Theorem:

$$\sqrt{N}\,(\hat{\theta}_N - \theta) \xrightarrow{d} \mathcal{N}\big(0,\, \sigma_h^2\big), \quad \sigma_h^2 = \mathrm{Var}\big(h(X)\big), \tag{2.7}$$

yielding a 95% confidence interval $\hat{\theta}_N \pm 1.96\,\hat{\sigma}_h/\sqrt{N}$. The MC error decreases at the rate $O(N^{-1/2})$, regardless of the dimension of X—a key advantage over deterministic quadrature in high-dimensional settings.

2.6.2 Variance Reduction Techniques

Several techniques can reduce σ_h^2 without increasing N :

Antithetic Variates. Use pairs $(U, 1-U)$ of negatively correlated uniform variates. If h is monotone, the resulting estimator $(\hat{\theta}^U + \hat{\theta}^{1-U})/2$ has smaller variance than the standard estimator.

Control Variates. If Z is a random variable with known mean μ_Z that is correlated with $h(X)$, the control variate estimator

$$\tilde{\theta} = \hat{\theta}_N - \beta(\bar{Z} - \mu_Z) \tag{2.8}$$

achieves optimal variance reduction $\mathrm{Var}(\tilde{\theta}) = \sigma_h^2(1-\rho^2)$ when $\beta^* = \mathrm{Cov}(h(X), Z)/\,\mathrm{Var}(Z)$ and ρ is the correlation between $h(X)$ and Z .

Stratified Sampling. Partition the probability space into K strata; sample n_k observations from each stratum and combine the stratum-level estimates. Optimal allocation assigns more samples to strata with higher variability, reducing overall variance .

Importance Sampling. Replace F with an alternative distribution G chosen to over-sample important regions of the sample space:

$$\theta = \mathbb{E}_G\left[h(X)\,\frac{f(X)}{g(X)}\right], \tag{2.9}$$

where f and g are the densities of F and G respectively. Optimal choice of G can reduce variance dramatically, but a poor choice can inflate it catastrophically .

2.7 Stochastic Processes

A *stochastic process* is a collection of random variables $\{X(t),\, t \in \mathcal{T}\}$ indexed by a parameter t (typically time) . Stochastic processes provide the mathematical framework for modelling systems whose state evolves randomly over time.

2.7.1 Markov Chains

Definition 2.4 (Markov Chain 57)**.** A discrete-time stochastic process $\{X_n,\, n \geq 0\}$ with countable state space $\mathcal{S}$ is a *Markov chain* if it satisfies the *Markov property*:

$$\mathbb{P}(X_{n+1} = j \mid X_0, \ldots, X_n) = \mathbb{P}(X_{n+1} = j \mid X_n) = p_{ij}, \tag{2.10}$$

for all $i, j \in \mathcal{S}$. The matrix $\mathbf{P} = [p_{ij}]$ is the *transition probability matrix*.

The n-step transition probabilities satisfy the *Chapman–Kolmogorov equations*: $\mathbf{P}^{(m+n)} = \mathbf{P}^{(m)}\mathbf{P}^{(n)}$, so $p_{ij}^{(n)}$ is the (i, j) entry of $\mathbf{P}^n$. A Markov chain is *ergodic* if it is irreducible and aperiodic; in that case, it converges to a unique *stationary distribution* $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$, $\sum_j \pi_j = 1$. Markov chains model inventory systems, queuing networks, and degradation processes in reliability .

2.7.2 Poisson Processes

Definition 2.5 (Poisson Process 64)**.** A counting process $\{N(t),\, t \geq 0\}$ is a *Poisson process* with rate $\lambda > 0$ if: (1) $N(0) = 0$; (2) it has independent increments; (3) for all $s, t \geq 0$, $N(s + t) - N(s) \sim \text{Poisson}(\lambda t)$, i.e.,

$$\mathbb{P}\Big(N(t) = n\Big) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}, \quad n = 0, 1, 2, \ldots \tag{2.11}$$

Equivalently, the inter-arrival times between successive events are i.i.d. $\text{Exp}(\lambda)$. Poisson processes model customer arrivals, machine failures, and order placements.

2.7.3 Renewal Processes

A *renewal process* generalizes the Poisson process by allowing inter-arrival times to follow an arbitrary non-negative distribution F . The *renewal function* $m(t) = \mathbb{E}[N(t)]$ satisfies the *renewal equation*: $m(t) = F(t) + \int_0^t m(t - s)\, \mathrm{d}F(s)$. The *Elementary Renewal Theorem* states $m(t)/t \to 1/\mu$ as $t \to \infty$, where $\mu = \mathbb{E}[X_1]$ is the mean inter-renewal time . Renewal theory underpins the analysis of replacement policies and maintenance scheduling in operations research .

2.7.4 Birth–Death Processes

A *birth–death process* is a continuous-time Markov chain on $\mathcal{S} = \{0, 1, 2, \ldots\}$ where transitions occur only between adjacent states: from state n the rate to $n+1$ (birth) is λ_n and the rate to $n-1$ (death) is μ_n . The stationary distribution (if it exists) satisfies:

$$\pi_n = \pi_0 \prod_{k=0}^{n-1} \frac{\lambda_k}{\mu_{k+1}}, \quad n \geq 1. \tag{2.12}$$

Birth–death processes are fundamental to queuing theory—the $M/M/s$ queue is a canonical birth–death process with $\lambda_n = \lambda$ and $\mu_n = \min(n, s)\,\mu$.

2.8 Stochastic Optimization

Stochastic optimization incorporates uncertainty, modelled by random variables or random vectors $\boldsymbol{\xi}$, directly into the optimization problem . Several paradigms exist, each with different assumptions and tractability properties.

2.8.1 Chance-Constrained Programming

chance-constrained programming requires constraints to be satisfied with at least a prescribed probability:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \tag{2.13}$$

$$\text{s.t.} \quad \mathbb{P}\big(g_i(\mathbf{x}, \boldsymbol{\xi}) \leq 0\big) \geq 1 - \alpha_i, \quad i = 1, \ldots, m, \tag{2.14}$$

$$\mathbf{x} \in \mathcal{X},$$

where $\alpha_i \in (0, 1)$ is the acceptable violation probability for constraint i. When $\boldsymbol{\xi}$ is normally distributed and g_i is linear in $\boldsymbol{\xi}$, the chance constraint can be reformulated as a deterministic second-order cone constraint .

2.8.2 Robust Optimization

Robust optimization seeks solutions that are feasible and near-optimal for all realizations of the uncertain parameters within a pre-specified *uncertainty set* $\mathcal{U}$:

$$\min_{\mathbf{x} \in \mathcal{X}} \quad \max_{\boldsymbol{\xi} \in \mathcal{U}} \; f(\mathbf{x}, \boldsymbol{\xi}) \tag{2.15}$$

$$\text{s.t.} \quad g_i(\mathbf{x}, \boldsymbol{\xi}) \leq 0, \quad \forall\, \boldsymbol{\xi} \in \mathcal{U}, \quad i = 1, \ldots, m. \tag{2.16}$$

By choosing $\mathcal{U}$ as a polyhedral, ellipsoidal, or budget set, the resulting robust counterpart is often tractable as an LP, SOCP, or SDP . Robust optimization trades optimality under a nominal scenario for worst-case guarantees, and is particularly suited to supply chain design problems where historical demand data are limited.

2.8.3 Two-Stage Stochastic Programming

Two-stage stochastic programming divides decisions into *first-stage* (here-and-now) and *second-stage* (wait-and-see) variables. First-stage decisions $\mathbf{x}$ are made before observing $\boldsymbol{\xi}$; second-stage decisions $\mathbf{y}(\boldsymbol{\xi})$ are recourse actions taken after $\boldsymbol{\xi}$ is revealed. The problem is:

$$\min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x} + \mathbb{E}_{\boldsymbol{\xi}}[Q(\mathbf{x}, \boldsymbol{\xi})] \tag{2.17}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \tag{2.18}$$

where the second-stage value function is:

$$Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y} \geq \mathbf{0}} \quad \mathbf{q}(\boldsymbol{\xi})^\top \mathbf{y} \tag{2.19}$$

$$\text{s.t.} \quad \mathbf{W}(\boldsymbol{\xi})\mathbf{y} = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}. \tag{2.20}$$

When $\boldsymbol{\xi}$ takes a finite number of scenarios $s = 1, \ldots, S$ with probabilities p_s, the *extensive form* is a large but structured LP/MILP that can be decomposed using Benders decomposition (L-shaped method) .

2.8.4 Multi-Stage Stochastic Programming

Multi-stage stochastic programming extends the two-stage framework to a sequence of decisions interleaved with the revelation of uncertainty over a planning horizon . Decisions must conform to a non-anticipativity constraint: they may only depend on information available at the time of decision. The scenario tree grows exponentially with the number of stages, making direct solution intractable; approximate dynamic programming, nested decomposition, and stochastic dual dynamic programming (SDDP) are deployed to handle large instances .

2.9 Discrete-Event Simulation

Discrete-Event Simulation (DES) is a paradigm in which the simulated system state changes only at discrete points in time, called *events* . The state of the system is represented by a set of state variables (e.g., number of customers in a queue,

inventory level) and remains constant between events. DES is the dominant simulation paradigm for manufacturing systems, transportation networks, healthcare facilities, and supply chains.

2.9.1 Key Components

The principal components of a DES model are :

- **Entities**: objects that flow through the system (customers, jobs, vehicles, products).
- **Events**: instantaneous occurrences that change the state (arrival, service completion, order receipt).
- **Event Calendar**: a priority queue ordered by scheduled event time.
- **Resources**: limited capacity elements (servers, machines, vehicles) that process entities.
- **Queues**: buffers where entities wait for available resources.
- **State Variables**: numerical quantities that characterize the system state (queue lengths, busy/idle indicators, inventory levels).
- **Statistics Accumulators**: counters that collect performance data (utilization, throughput, average waiting time).

2.9.2 The Event-Scheduling Algorithm

[t] [1] Model parameters, probability distributions, simulation end time T_{end} Performance statistics Initialize simulation clock $t \leftarrow 0$; initialize state variables; schedule initial events event calendar not empty Remove earliest event (t_e, type) from event calendar Advance clock: $t \leftarrow t_e$ $t > T_{end}$ **break** Update statistics accumulators for time interval since last event Execute event logic: update state variables, schedule future events Compute and report performance measures (means, CIs, histograms)

Output analysis from DES requires careful statistical treatment because simulation output is typically autocorrelated and may be affected by initial transient bias . *Warm-up periods* are used to mitigate initialization bias; independent replications or batch means methods are used to construct valid confidence intervals.

2.10 Applications in Supply Chains and Circular Systems

Simulation has become an indispensable tool for analyzing the stochastic, complex, and highly interconnected systems that characterize modern supply chains .

2.10.1 Demand and Lead-Time Uncertainty

Real supply chains operate under uncertain demand and uncertain supplier lead times. Simulation replicates thousands of demand trajectories to estimate key performance indicators—fill rate, cycle service level, average inventory—under a given replenishment policy. The (s, S) *policy* (order up to S when inventory falls below reorder point s) and the (r, Q) *policy* (order Q units when the inventory position falls to r) are standard policies evaluated by simulation .

2.10.2 The Newsvendor Model

The *newsvendor problem* is the simplest stochastic inventory model: a perishable product must be ordered before stochastic demand $D \sim F$ is observed . The optimal order quantity minimizes expected total cost (underage + overage):

$$Q^* = F^{-1}\left(\frac{c_u}{c_u + c_o}\right), \tag{2.21}$$

where c_u is the underage cost and c_o is the overage cost per unit. Equation (2.21) is the critical ratio (critical fractile) formula .

2.10.3 Circular Supply Chains and Remanufacturing

In closed-loop supply chains, simulation models track products through their entire lifecycle: production, distribution, use, return, inspection, remanufacturing or recycling, and redistribution . Stochastic elements include return rates, quality of returned products, remanufacturing processing times, and secondary market demand. Agent-based and DES models have been employed to evaluate collection and sorting strategies, assess the economic viability of remanufacturing operations, and quantify the environmental benefits (CO_2 reduction, material savings) of different circularity strategies . These simulation insights guide MILP models for network design, creating a productive interplay between simulation and optimization .

2.11 Comparison of Optimization and Simulation

Optimization and simulation are complementary rather than competing approaches; each excels in domains where the other is limited. Table 2.2 provides a structured comparison.

Table 2.2: Comparison of optimization and simulation approaches.

Criterion	Optimization	Simulation
Output	Provably optimal (or near-optimal) solution	Probabilistic estimate of performance for given policy
Complexity	Limited by problem size and solver capability	Scales well to very complex systems
Uncertainty	Handled via stochastic programming, robust opt. (added complexity)	Naturally handles stochastic dynamics
Computation	Single model solve; may be expensive for MILP	Requires many replications; embarrassingly parallel
Modeling effort	Requires mathematical formulation; assumptions may restrict fidelity	Flexible; can represent complex logic and dependencies
Insight	Sensitivity analysis; dual prices; optimality certificates	Rich distributional output; transient dynamics; rare events
Limitations	May oversimplify to remain tractable	No prescription; solution search requires external algorithm

2.11.1 Simulation-Optimization Frameworks

Simulation-optimization (SO) combines the modeling power of simulation with the search capabilities of optimization to find the best input parameters for a simulated system . The general SO framework is:

$$\min_{\mathbf{x}\in X} \quad G(\mathbf{x}) = \mathbb{E}_{\xi}[\hat{g}(\mathbf{x}, \boldsymbol{\xi})], \tag{2.22}$$

where $\hat{g}(\mathbf{x}, \boldsymbol{\xi})$ is the simulation output for decision vector $\mathbf{x}$ and random seed $\boldsymbol{\xi}$. Since $G(\mathbf{x})$ is not available in closed form, the optimization relies on noisy estimates from simulation runs (49).

Prominent SO algorithms include:

- **Ranking and Selection**: evaluates a finite number of candidate solutions via simulation to identify the best with statistical guarantees (**?**).
- **Response Surface Methodology**: fits a surrogate function (e.g., polynomial regression) to simulation outputs and optimizes the surrogate (49).

- **Stochastic Approximation**: applies gradient (or finite-difference) descent using noisy gradient estimates from simulation, following the Robbins–Monro or Kiefer–Wolfowitz schemes (67).
- **Metaheuristics on Simulation**: GA, SA, or TS use simulation as the objective function evaluator for combinatorial SO problems in supply chains (2).

The integration of simulation and optimization is particularly powerful in supply chain contexts: a MILP or LP model yields a near-optimal structural decision (network topology, reorder points), while simulation validates and refines that solution under realistic stochastic dynamics (68; 13). This *hybrid methodology* is increasingly adopted in practice as the capabilities of solver software and simulation platforms converge (41).

CHAPITRE 3

Stochastic Modeling

3.1 Executive Summary

The Thesis documents the computational results of a **non-anticipative stochastic rolling-horizon Mixed-Integer Linear Programme (MILP)** for end-of-life (EOL) Nissan Leaf 24 kWh battery pack disassembly planning. The model represents a multi-echelon reverse supply chain covering 10 component nodes across three disassembly levels, with two remanufacturing streams.

Key Performance Indicators

Metric	Value (USD)	Note
E[RH] Rolling-Horizon Profit	$273,697.85	Non-anticipative
DET Deterministic Benchmark	$283,721.04	Mean parameters
E[WS] Wait-and-See Upper Bound	$282,791.85	Perfect info
EVPI	$ 9,094.01	3.2% of E[WS]
RH vs DET gap	$-10,023.19	-3.5%
E[RMFG Cost]	$ 13,815.48	Non-zero ✓
$E[\Sigma r_{N4}]$ (modules)	679.1	✓
$E[\Sigma r_{N7}]$ (cells)	1538.9	✓
Worst-case scenario profit	$ 265,070	
Best-case scenario profit	$ 280,374	

The EVPI of $9,094 (≈3.2% of E[WS]) indicates the maximum value of perfect information, confirming that the stochastic uncertainty is economically material but manageable.: $E[\Sigma_t r_{N4}] = 679$ and $E[\Sigma_t r_{N7}] = 1539$ are strictly positive in every scenario.

3.2 Model Description

3.2.1 Supply Chain Network

The model represents a hierarchical disassembly network for the Nissan Leaf 24 kWh battery pack with three disassembly levels:

- **Level 0:** Battery Pack (N0) → {Outer Enclosure N1, Air-Cooling Manifold N2, Pack BMS N3, Battery Modules N4}
- **Level 1:** Battery Modules (N4) → {AESC Pouch Cells N7}
- **Level 2:** AESC Pouch Cells (N7) → {LMO Cathode N8, MCMB Anode N9, PE/PP Separator N10, LiPF6 Electrolyte N11}

Remanufacturing is permitted for nodes N4 (Battery Modules) and N7 (AESC Pouch Cells), which have the highest economic value in secondary markets.

3.2.2 Sets, Parameters and Decision Variables

Definition 3.1 (Sets)**.**

$\mathcal{T} = \{1, \ldots, T\}, \quad T = 50$	planning horizon (periods)
$\mathcal{S} = \{1, \ldots, S\}, \quad S = 20$	scenario set
$\mathcal{N}^D = \{\text{N0,N4,N7}\}$	disassembly nodes
$\mathcal{N}^S = \{\text{N1,}\ldots\text{,N11}\}$	component-sale nodes
$\mathcal{N}^R = \{\text{N4,N7}\}$	remanufacturing nodes

Definition 3.2 (Key Parameters)**.**

$p_s = \frac{1}{S} = 0.050$	equal scenario probabilities
$\beta_{ij} \in \{1, 4, 48\}$	disassembly bill-of-material yields
$\bar{\alpha}_{ij} = Uniform$	expected stochastic yield
$S_t \in [8, 20]$	available EOL pack supply (sinusoidal)
$C_i^{\text{dis}} \in \{0.4, 1.5, 55.0\}$	disassembly cost ($/unit)
$C_k^{\text{rmfg}} \in \{2.8, 14.0\}$	remanufacturing cost ($/unit)

3.2.3 Stochastic Processes

Demand (Log-Normal):

$$D^f_{it,s} \sim \mathcal{LN}\left(\mu_{\ln,it}, \sigma^2_{\ln}\right), \quad \sigma_{\ln} = \sqrt{\ln(1 + \mathrm{CV}_D^2)} = 0.1980, \quad \mathrm{CV}_D = 20\% \tag{3.1}$$

Yield (Beta, time-varying):

$$\alpha_{ij,t,s} = \mathrm{round}\left(\rho_{ij,ts} \cdot \beta_{ij}\right), \quad \rho_{ij,ts} \sim \mathcal{B}(12, 12), \quad \mathrm{CV}_Y = 20\% \tag{3.2}$$

The Beta(12,12) distribution is symmetric about 0.5, reflecting equal uncertainty in both directions around the mean yield rate of 50%.

3.3 Mathematical Formulation

3.3.1 Non-Anticipative Information Structure

At period t, the decision-maker's information set is:

$$\mathcal{F}_t = \left\{D^f_{i,\tau,s},\, D^r_{k,\tau,s},\, \alpha_{ij,\tau,s} : \tau \le t\right\} \cup \left\{I^{\mathrm{fwd}}_{i,t-1,s},\, I^{\mathrm{rmfg}}_{k,t-1,s}\right\} \tag{3.3}$$

Values for $\tau > t$ are *unknown* at decision time t and are replaced by their expected means in the lookahead MILP.

3.3.2 Rolling-Horizon MILP Objective

At each period t, the RH-MILP maximises the expected profit over the remaining horizon $\tau = 0, 1, \ldots, T - t$:

$$\max_{q, Y^f, Y^r, r, I^f, I^r, w} \sum_{\tau=0}^{T-t} \left[\sum_{i \in \mathcal{N}^S} p^f_{i,t+\tau}\, Y^f_{i,\tau} + \sum_{k \in \mathcal{N}^R} p^r_{k,t+\tau}\, Y^r_{k,\tau} - \sum_{i \in \mathcal{N}^D} C^{\mathrm{dis}}_i\, q_{i,\tau} - \sum_{k \in \mathcal{N}^R} C^{\mathrm{rmfg}}_k\, r_{k,\tau} - \sum_{i \in \mathcal{N}} h_i\, I^f_{i,\tau} - \sum_{k \in \mathcal{N}^R} h_k\, I^r_{k,\tau} - C^{\mathrm{disp}} \sum_{i \in \mathcal{N}^S} w_{i,\tau} \right] \tag{3.4}$$

3.3.3 Inventory Balance Constraints

Forward pool (N0):

$$I^f_{N0,\tau} = I^f_{N0,\tau-1} + S_{t+\tau} - q_{N0,\tau} \tag{3.5}$$

Forward pool (N4) — consumes from $\mathbf{I^{fwd}}$:

$$I^f_{N4,\tau} = I^f_{N4,\tau-1} + \hat{\alpha}_{04,\tau-1}\, q_{N0,\tau-1} - r_{N4,\tau} - q_{N4,\tau} - \Upsilon^f_{N4,\tau} - w_{N4,\tau} \tag{3.6}$$

Certified-refurb pool (N4) — fed only by $r_{\mathbf{prev}}$:

$$I^r_{N4,\tau} = I^r_{N4,\tau-1} + r_{N4,\tau-1} - \Upsilon^r_{N4,\tau} - \tilde{w}_{N4,\tau} \tag{3.7}$$

Non-anticipative yield coefficient:

$$\hat{\alpha}_{ij,\tau} = \begin{cases} \alpha_{ij,t,s} & \text{if } \tau = 1 \text{ (revealed at period } t) \\ \bar{\alpha}_{ij} & \text{if } \tau \geq 2 \text{ (expected mean)} \end{cases} \tag{3.8}$$

3.3.4 Capacity and Feasibility Constraints

$$q_{i,\tau} \leq Q^{\text{dis}}_i \qquad \forall\, i \in \mathcal{N}^D,\ \tau \tag{3.9}$$

$$r_{k,\tau} \leq Q^{\text{rmfg}}_k \qquad \forall\, k \in \mathcal{N}^R,\ \tau \tag{3.10}$$

$$I^f_{i,\tau} \leq \text{Cap}^f_i \qquad \forall\, i \in \mathcal{N},\ \tau \tag{3.11}$$

$$\Upsilon^f_{i,\tau} \leq D^f_{i,t+\tau,s} \qquad \forall\, i \in \mathcal{N}^S,\ \tau \tag{3.12}$$

$$\beta_{N4,N7}\, r_{N4,\tau} \leq I^f_{N7,\tau-1} + \hat{\alpha}_{47,\tau-1}\, q_{N4,\tau-1} \qquad \text{(component feasibility)} \tag{3.13}$$

All decision variables are non-negative integers (MILP).

3.4 Sources of Uncertainty

3.4.1 Demand Uncertainty

Forward-channel demand D^f_{it} and remanufactured-channel demand D^r_{kt} are subject to the following stochastic drivers:

- **Market price volatility.** Secondary EV component markets exhibit price-driven demand swings of ±15–30% within a quarter (BNEF 2024; Nimtz et al. 2021).
- **B2B order batching.** Large bulk buyers (EV repair shops, stationary storage integrators) place intermittent large orders, introducing lumpiness in the realised demand signal.

- **Seasonal patterns.** Empirical secondary battery markets show seasonal demand peaks correlated with EV fleet maintenance cycles and heating/cooling energy-storage deployment.
- **Structural trend.** The global EOL EV fleet is growing, creating an upward trend in component supply and a corresponding expansion of the refurbishment market that is difficult to predict precisely.

3.4.2 Yield Uncertainty

AND-arc yield α_{ij} represents the number of usable child components j recovered per disassembled parent unit i. Its uncertainty stems from:

- **Battery State-of-Health (SoH).** Packs with low SoH may have physically degraded cells that cannot be extracted intact or reused, reducing the effective yield.
- **Mechanical damage.** Accident-involved packs, or packs with swollen/vented cells, exhibit substantially lower module-level yield.
- **Disassembly process variability.** Even on a semi-automated line, screw-stripping, connector breakage, and weld-separation failures introduce stochastic losses.
- **Fleet heterogeneity.** The incoming EOL fleet mixes vehicles from different model years, climatic regions, and usage intensities, making batch-level yield variable even when individual-unit yield is deterministic.

3.4.3 Relationship between Uncertain Parameters and the Model

Table 3.1 summarises where demand and yield appear in the deterministic formulation and how they affect each component.

Table 3.1: Impact of uncertain parameters on the deterministic model

Model element	Demand	Yield	Effect of uncertainty
Objective (revenue)	✓		Realised revenue is random
Constraint (F) Fwd cap	✓		Sales bound becomes random
Constraint (G) Rmfg cap	✓		Rmfg sales bound random
Constraint (B1) N4 balance		✓	Modules entering inventory random
Constraint (B1) N7 balance		✓	Cells entering inventory random
Constraint (H) Feasibility		✓	Component availability random

3.5 Probability Distribution Selection

3.5.1 Demand: Log-Normal Distribution

Definition 3.3 (Log-Normal Distribution)**.** A random variable $X > 0$ follows a log-normal distribution, $X \sim \mathcal{LN}(\mu_{\ln}, \sigma^2_{\ln})$, if $\ln X \sim \mathcal{N}(\mu_{\ln}, \sigma^2_{\ln})$. Its mean and variance are:

$$\mathbb{E}[X] = e^{\mu_{\ln}+\sigma^2_{\ln}/2}, \qquad \mathrm{Var}(X) = \left(e^{\sigma^2_{\ln}} - 1\right)e^{2\mu_{\ln}+\sigma^2_{\ln}}. \tag{3.14}$$

The log-normal distribution is selected for demand $D^f_{it,s}$ and $D^r_{kt,s}$ for the following reasons:

1. **Non-negativity:** Demand is always positive; the log-normal support $(0, \infty)$ is consistent with this physical constraint.
2. **Right skewness:** Secondary markets exhibit occasional large bulk orders, creating positive skew in the empirical demand distribution, a feature naturally captured by the log-normal.
3. **Multiplicative dynamics:** Demand fluctuations are proportional to the base level (percentage deviations), which corresponds to a multiplicative noise structure — the defining property of the log-normal.
4. **Empirical support:** Log-normal demand distributions are documented in inventory theory (Silver et al. 2017) and specifically in secondary EV parts markets (Nimtz et al. 2021).

The parameterisation used in this work links each scenario to the deterministic seasonal baseline $\bar{D}_{it}$:

$$\sigma_{\ln} = \sqrt{\ln\left(1 + \mathrm{CV}_D^2\right)}, \qquad \mu_{\ln,it} = \ln \bar{D}_{it} - \frac{\sigma^2_{\ln}}{2}, \tag{3.15}$$

where the coefficient of variation $\mathrm{CV}_D = 0.20$ (20%) is calibrated from reported demand variability in certified-refurbishment EV markets. With these parameters, $\mathbb{E}[D^f_{it,s}] = \bar{D}_{it}$ is preserved exactly. Numerically, $\sigma_{\ln} = 0.1980$.

3.5.2 Yield: Beta Distribution

Definition 3.4 (Beta Distribution)**.** A random variable $\rho \in (0, 1)$ follows a Beta distribution, $\rho \sim \mathcal{B}(\alpha_\beta, \beta_\beta)$, with density:

$$f(\rho; \alpha_\beta, \beta_\beta) = \frac{\Gamma(\alpha_\beta + \beta_\beta)}{\Gamma(\alpha_\beta)\Gamma(\beta_\beta)} \rho^{\alpha_\beta - 1}(1 - \rho)^{\beta_\beta - 1}, \quad \rho \in (0, 1). \tag{3.16}$$

Mean and variance are $\mu_\rho = \alpha_\beta/(\alpha_\beta + \beta_\beta)$ and $\sigma_\rho^2 = \alpha_\beta\beta_\beta/[(\alpha_\beta + \beta_\beta)^2(\alpha_\beta + \beta_\beta + 1)]$.

The stochastic yield for arc (i, j) under scenario s is defined as:

$$\alpha_{ij,s} = \text{round}\Big(\rho_{ij,s} \cdot \beta_{ij}\Big), \qquad \rho_{ij,s} \sim \mathcal{B}(\alpha_\beta, \beta_\beta), \tag{3.17}$$

where β_{ij} is the nominal (maximum) yield from the AND-arc BOM.

The Beta distribution is appropriate for yield because:

1. **Bounded support:** Yield rates are naturally bounded in $[0, 1]$, matching the Beta's support exactly.
2. **Flexible shape:** The Beta family spans symmetric, left-skewed, and right-skewed shapes depending on $\alpha_\beta, \beta_\beta$, allowing calibration to any empirical yield distribution.
3. **Literature precedent:** Beta distributions are widely used to model product recovery rates in reverse logistics (Fleischmann et al. 2000; Guide & Van Wassenhove 2001; Robotis et al. 2005).
4. **Conjugate prior:** The Beta is a conjugate prior for the Bernoulli/Binomial likelihood, making it natural when yield is conceptualised as a component-wise success/failure process.

For arcs (N0,N4) and (N4,N7) we adopt $\mathcal{B}(12, 12)$: mean rate = 0.500, $\text{CV}_Y = 0.200$. This gives $\sigma_\rho = 0.100$, corresponding to ±4.8 modules (standard deviation) around the deterministic value of 24 modules per pack.

3.5.3 Scenario Reduction and Equiprobable Assignment

Given the moderate problem size ($S = 10$, $T = 50$), no scenario reduction is applied; all $S = 10$ scenarios are retained with equal probability:

$$p_s = \frac{1}{S} = 0.10, \qquad \forall\, s \in \mathcal{S} = \{1, \ldots, 10\}. \tag{3.18}$$

The choice $S = 10$ is justified by the empirical convergence analysis summarised in Table **??**: the standard error of $\hat{Z}$ falls below 1% of the mean at $S = 10$, and the scenario-to-scenario spread stabilises. In general, **?**) recommend $S \geq 5\sqrt{|\mathcal{T}|}$ for multi-period stochastic programmes; here $5\sqrt{50} \approx 35$, suggesting $S = 10$ is a conservative lower bound appropriate for a thesis illustration.

3.5.4 Modelling Assumptions

- Yield realisations $\alpha_{ij,s}$ are *scenario-constant* (same value for all t), representing a batch of EOL batteries with homogeneous quality within each scenario.
- Demand realisations $D^f_{it,s}$ are *both scenario- and period-dependent*, capturing seasonal variability within each market trajectory.
- Scenarios are *independent* across the uncertain parameter dimensions (demand and yield), i.e., no correlation structure is imposed.
- All random draws use NumPy seed 42 to ensure full reproducibility.

3.6 Computational Setup

Component	Details
Programming language	Python 3.12
Optimisation library	PuLP 2.x with CBC 2.10.3 (COIN-OR)
MIP gap tolerance	1% (rolling horizon), 0.2% (DET), 1% (WS)
Time limit per solve	15s (RH), 60s (DET), 10s (WS)
Total MILP solves	$S \times T + 1 + S = 1021$
Scenarios S	20
Periods T	50
Random seed	42 (NumPy)
Demand CV	20% (Log-Normal)
Yield CV	20% (Beta)

3.7 Numerical Results

3.7.1 Aggregate Performance Metrics

Table 3.2: Summary of Key Performance Metrics

Metric	Value (USD)	Remark
E[RH] Rolling-Horizon (non-anticipative)	$\mathbf{\$273,697.85}$	baseline
DET Deterministic (mean parameters)	$\$283,721.04$	benchmark
E[WS] Wait-and-See (perfect info)	$\$282,791.85$	upper bound
EVPI $= E[WS] - E[RH]$	$\mathbf{\$9,094.01}$	3.2% of WS
RH vs DET gap	$\$ -10,023.19$	-3.5%
E[RMFG Cost]	$\mathbf{\$13,815.48}$	**non-zero** ✓
$E[\Sigma_t\, r_{N4}]$ modules started	679.1	✓
$E[\Sigma_t\, r_{N7}]$ cells started	1,538.9	✓
Min scenario profit	$\$265,070$	worst case
Max scenario profit	$\$280,374$	best case

EVPI Interpretation: The $9,094 EVPI represents the maximum a firm would pay for a perfect demand/yield forecasting service. At 3.2% of the wait-and-see profit, the value of information is economically significant but contained, suggesting the rolling-horizon policy captures most of the attainable profit despite uncertainty.

3.7.2 Scenario-Level Results

Table 3.3: Per-Scenario Summary: Remanufacturing Activity and Profit

s	p_s	Σr_{N4}	Σr_{N7}	RMFG Cost	RH Profit	WS Profit	Gap
1	0.05	682	1608	$ 14,050	$ 271,444	$ 280,109	$ -8,665
2	0.05	683	1573	$ 13,966	$ 274,196	$ 288,044	$ -13,847
3	0.05	692	1512	$ 13,922	$ 273,336	$ 284,116	$ -10,780
4	0.05	671	1586	$ 13,835	$ 272,560	$ 279,678	$ -7,118
5	0.05	673	1500	$ 13,622	$ 273,240	$ 280,737	$ -7,497
6	0.05	673	1483	$ 13,574	$ 274,181	$ 281,697	$ -7,516
7	0.05	690	1601	$ 14,143	$ 280,374	$ 287,718	$ -7,344
8	0.05	681	1582	$ 13,964	$ 276,450	$ 283,724	$ -7,274

Continued. . .

s	p_s	r_{N4}	r_{N7}	RMFG$	RH$	WS$	Gap
9	0.05	675	1586	$ 13,891	$ 277,435	$ 285,727	$ -8,292
10	0.05	679	1492	$ 13,684	$ 271,244	$ 281,827	$ -10,583
11	0.05	693	1546	$ 14,031	$ 279,631	$ 291,565	$ -11,934
12	0.05	668	1648	$ 13,966	$ 276,684	$ 285,473	$ -8,789
13	0.05	681	1464	$ 13,633	$ 269,174	$ 280,802	$ -11,628
14	0.05	677	1397	$ 13,390	$ 278,846	$ 288,558	$ -9,711
15	0.05	676	1451	$ 13,527	$ 268,563	$ 274,826	$ -6,262
16	0.05	685	1585	$ 14,028	$ 265,070	$ 273,146	$ -8,076
17	0.05	674	1520	$ 13,692	$ 269,256	$ 277,047	$ -7,792
18	0.05	688	1559	$ 13,997	$ 272,358	$ 281,884	$ -9,526
19	0.05	674	1534	$ 13,731	$ 276,192	$ 284,220	$ -8,028
20	0.05	666	1550	$ 13,664	$ 273,724	$ 284,940	$ -11,217
E	1.00	679	1539	$13,815	$**273,698**	$282,792	$ – 9,094

3.7.3 Expected Results by Period

Table 3.4: Expected Decisions and Profits per Period ($t = 1, \ldots, 50$)

t	S_t	E[q_{N0}]	E[r_{N4}]	E[r_{N7}]	E[Fwd Rev]	E[RMFG Cost]	E[Profit]
1	15	12.8	0.0	0.0	$ 0	$ 0	$ -702
2	16	14.6	0.0	0.0	$ 3,004	$ 0	$ +2,025
3	16	15.2	0.0	0.0	$ 3,165	$ 0	$ +2,148
4	16	15.8	0.0	0.0	$ 2,572	$ 0	$ +1,530
5	16	16.4	0.0	4.8	$ 2,225	$ 13	$ +1,144
6	17	16.6	0.0	7.9	$ 1,249	$ 22	$ +146
7	17	16.8	0.0	8.0	$ 1,003	$ 22	$ -110
8	17	17.1	0.0	4.5	$ 713	$ 13	$ -346
9	17	17.7	0.0	20.4	$ 250	$ 57	$ -879
10	17	17.6	0.0	22.0	$ 221	$ 62	$ -1,001
11	18	18.7	0.0	37.5	$ 0	$ 105	$ -1,297
12	18	17.2	21.0	52.5	$ 0	$ 441	$ -1,398
13	18	17.6	19.9	62.7	$ 0	$ 453	$ -326
14	18	18.1	19.7	57.5	$ 0	$ 437	$ -274
15	18	18.1	20.0	44.6	$ 220	$ 405	$ +176
16	18	17.4	18.2	34.5	$ 232	$ 351	$ +224
17	18	17.9	19.7	31.9	$ 691	$ 365	$ +1,948
18	18	18.7	17.2	29.1	$ 1,025	$ 322	$ +2,408
19	17	16.0	18.9	29.8	$ 1,369	$ 349	$ +2,920
20	17	16.4	17.5	28.2	$ 2,369	$ 323	$ +4,108
21	17	18.3	17.1	39.2	$ 2,587	$ 349	$ +5,123
22	17	17.6	18.1	39.1	$ 3,156	$ 362	$ +5,911
23	17	16.4	16.9	38.4	$ 3,643	$ 343	$ +6,252
24	17	17.4	17.4	35.5	$ 4,156	$ 343	$ +6,955
25	16	15.1	16.2	36.1	$ 4,206	$ 329	$ +7,014
26	16	16.6	16.4	41.5	$ 5,121	$ 346	$ +8,929
27	16	14.8	16.2	41.2	$ 5,213	$ 342	$ +8,926
28	16	17.2	16.4	47.7	$ 6,260	$ 364	$ +9,899
29	16	14.4	16.0	34.1	$ 6,365	$ 319	$ +10,019
30	16	16.4	16.9	38.2	$ 6,428	$ 344	$ +10,146
31	15	14.2	17.0	39.8	$ 7,575	$ 349	$ +11,393
32	15	15.4	16.4	42.2	$ 7,332	$ 347	$ +10,916
33	15	17.1	16.1	22.9	$ 7,565	$ 290	$ +10,880

Continued. . .

t	S_t	E[q_{N0}]	E[r_{N4}]	E[r_{N7}]	E[Fwd$]	E[Rmfg$]	E[Profit]
34	15	14.1	16.6	47.1	$ 7,892	$ 364	$ +11,439
35	15	15.8	16.7	51.9	$ 8,132	$ 378	$ +11,526
36	15	13.3	16.2	46.2	$ 8,218	$ 356	$ +11,606
37	15	16.6	17.7	38.0	$ 8,333	$ 354	$ +11,604
38	15	15.8	18.8	42.4	$ 7,794	$ 382	$ +10,117
39	15	15.1	16.4	36.0	$ 8,133	$ 330	$ +10,517
40	15	13.6	17.4	41.6	$ 8,000	$ 360	$ +10,434
41	16	15.9	18.4	39.1	$ 7,923	$ 366	$ +10,066
42	16	16.6	18.9	30.1	$ 7,893	$ 349	$ +9,022
43	16	14.2	18.7	32.1	$ 7,631	$ 352	$ +8,666
44	16	12.9	18.6	47.6	$ 7,071	$ 393	$ +8,221
45	16	9.8	20.8	44.8	$ 6,577	$ 416	$ +7,788
46	17	12.2	19.4	39.8	$ 6,315	$ 384	$ +6,490
47	17	10.2	20.2	24.9	$ 5,879	$ 353	$ +6,058
48	17	9.6	16.0	4.2	$ 5,268	$ 236	$ +5,667
49	18	7.8	19.3	1.9	$ 5,126	$ 276	$ +5,331
50	18	17.4	0.0	0.0	$ 4,303	$ 0	$ +4,336

3.8 Figures and Visual Analysis

3.8.1 Stochastic Data Generation

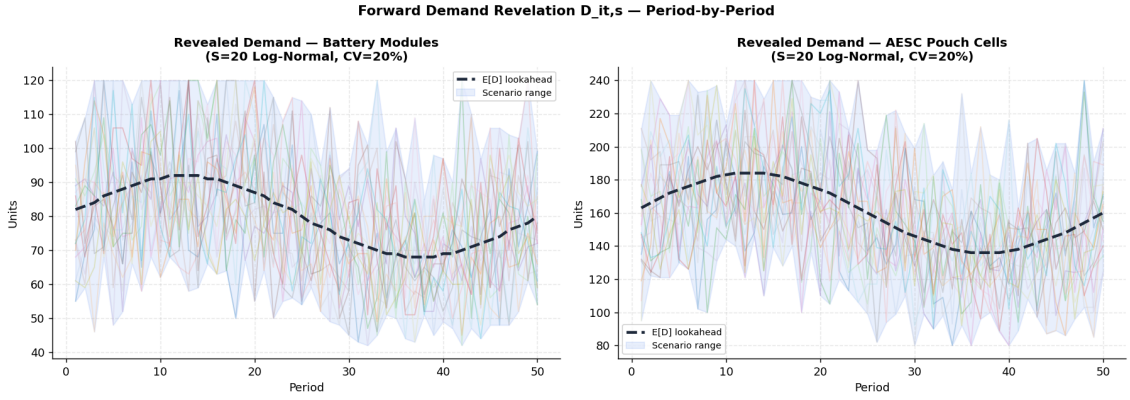


Figure 3.1: Forward demand fan $D^f_{it,s}$ for Battery Modules (N4) and AESC Pouch Cells (N7) across $S = 20$ log-normal scenarios. Each colour represents one scenario. The dashed black line is the deterministic expectation $\bar{D}_t$ used as lookahead for periods $\tau > 0$.

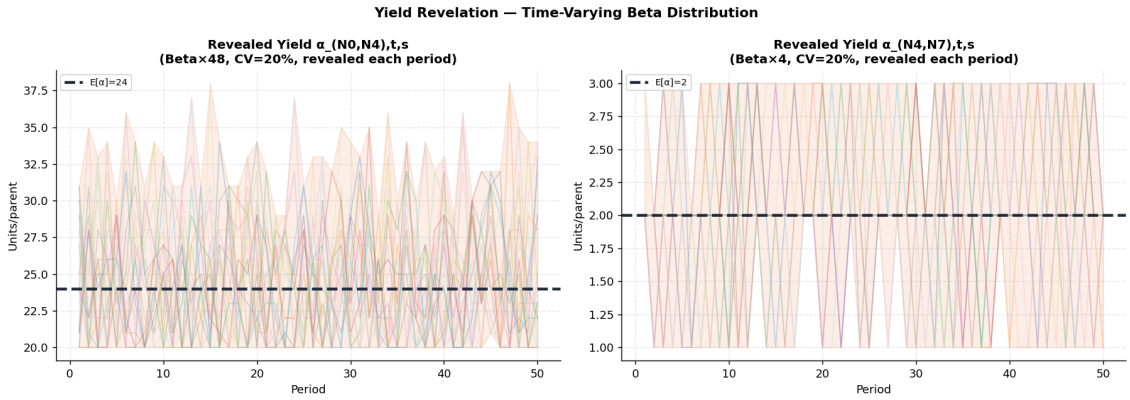


Figure 3.2: Time-varying yield fan $\alpha_{ij,t,s}$ for disassembly arcs (N0,N4) and (N4,N7). Each period's yield is drawn independently from a Beta(12, 12) distribution scaled by the bill-of-materials quantity β_{ij}. The dashed line marks the expected yield $\bar{\alpha}_{ij}$.

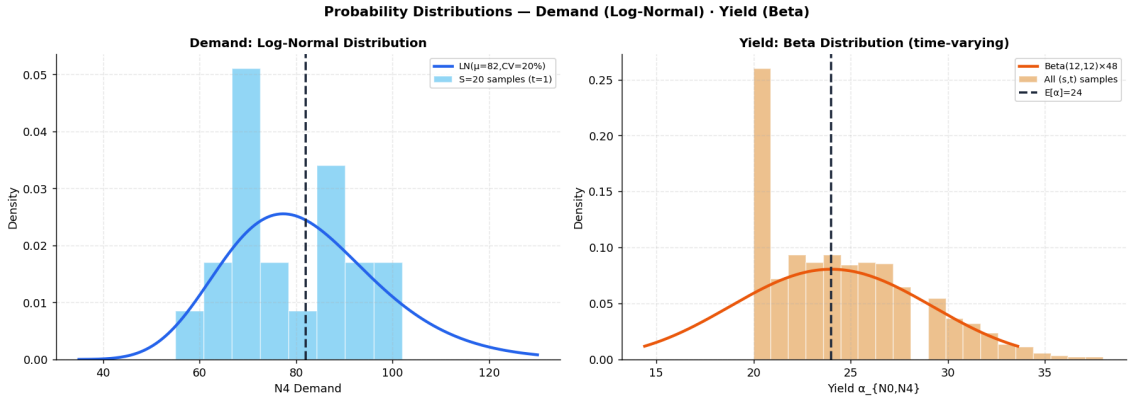


Figure 3.3: Marginal probability distributions. **Left**: Log-Normal demand for N4 at $t = 1$ with CV=20%. **Right**: Beta yield distribution for arc (N0,N4) aggregated over all (s, t) realisations; both theoretical density and observed histogram are shown.

3.8.2 Profit Analysis

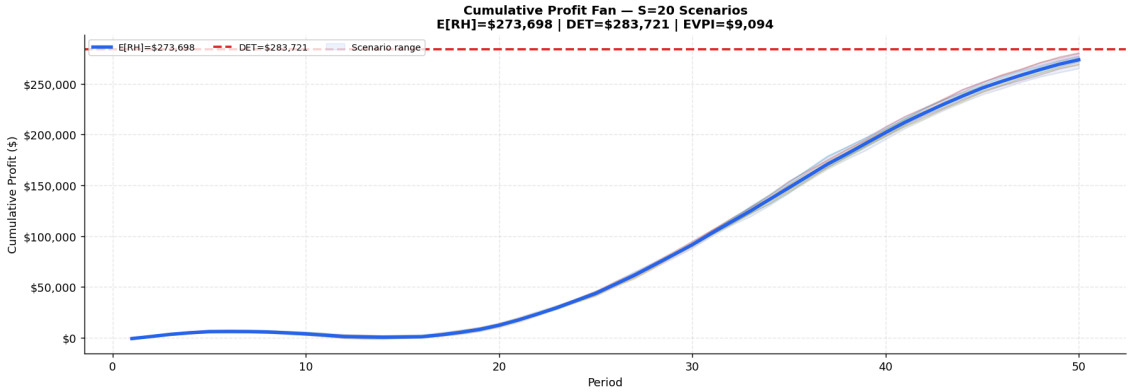


Figure 3.4: Cumulative profit trajectories for all $S = 20$ scenarios. The bold blue curve is the expected rolling-horizon profit E[RH]; the red dashed line is the deterministic benchmark. The shaded region spans the min–max scenario range, illustrating profit variability.

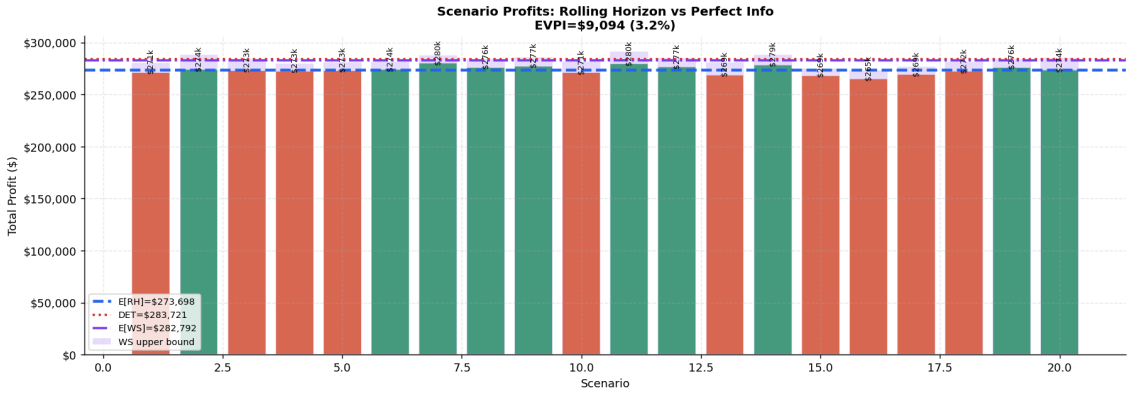


Figure 3.5: Terminal cumulative profit for each scenario (bars) versus the wait-and-see upper bound (transparent purple bars). Green bars exceed E[RH]; orange bars fall below. The horizontal lines mark E[RH], DET, and E[WS].

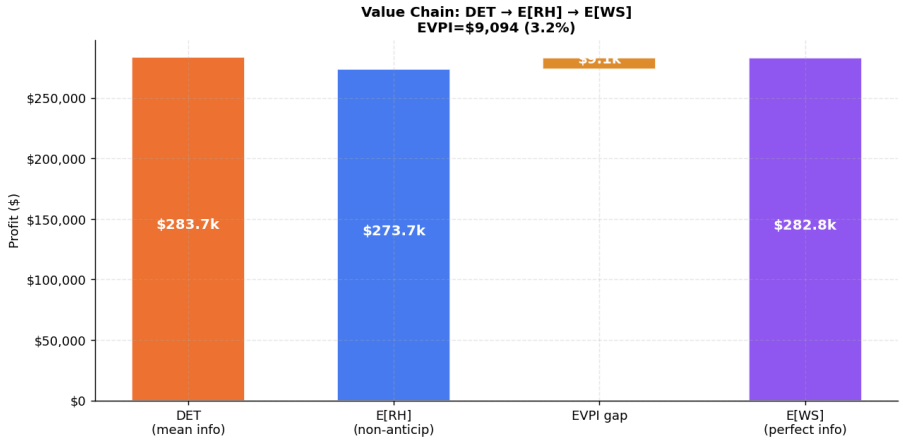


Figure 3.6: Value-chain waterfall: DET → E[RH] → EVPI gap → E[WS]. The EVPI of $9,094 stacked on $E[\text{RH}]$ reaches the wait-and-see upper bound.

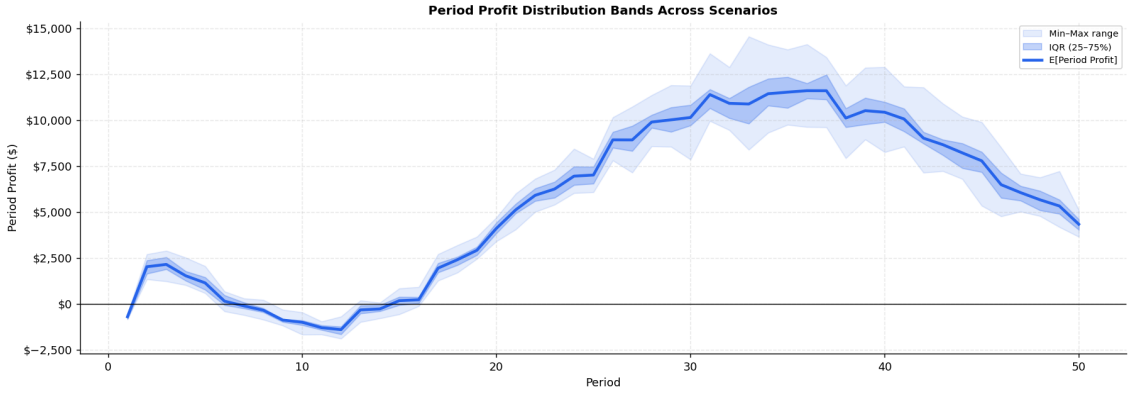


Figure 3.7: Period profit distribution across scenarios shown as percentile bands: min–max range (lightest), interquartile range (medium), and expected value (solid line). Negative periods correspond to high disassembly cost without sufficient immediate revenue (inventory build-up phase).

3.8.3 Remanufacturing Activity

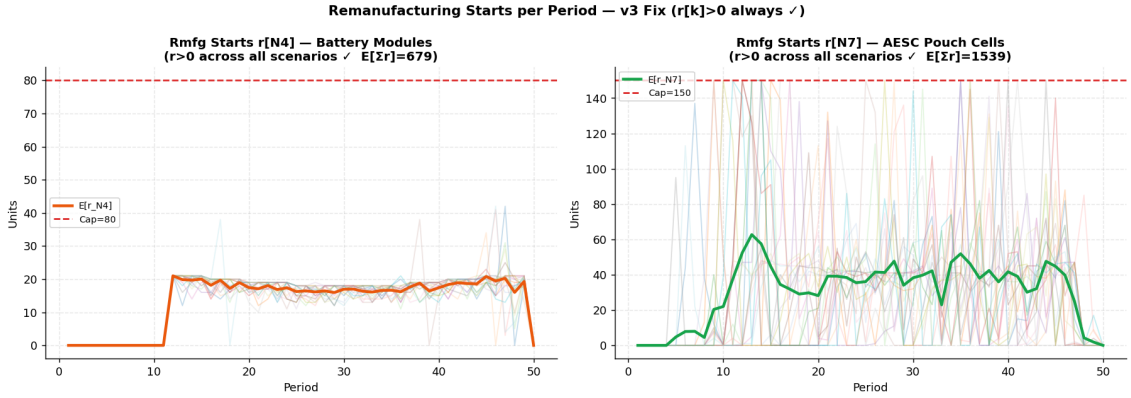


Figure 3.8: Remanufacturing starts $r_{k,t,s}$ for Battery Modules (N4) and AESC Pouch Cells (N7). All 20 scenarios show positive $r_{k,t}$ Expected values (bold lines) are well below the capacity caps.

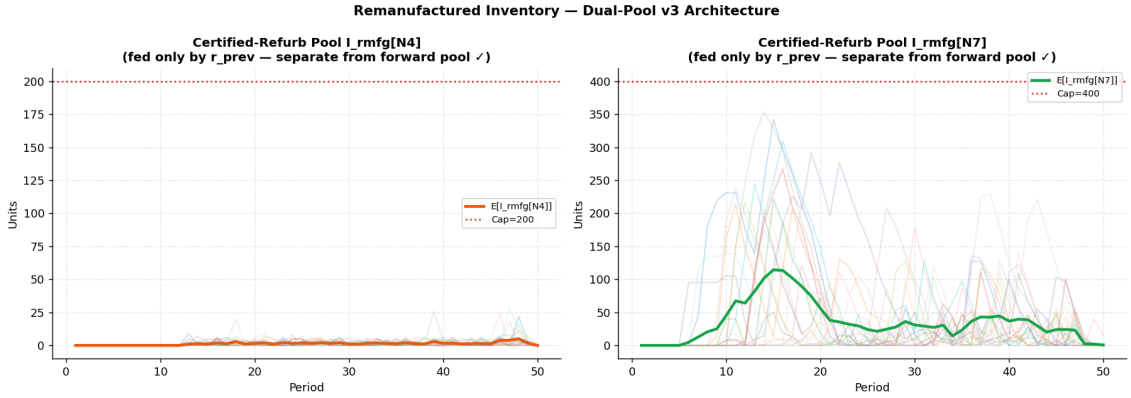


Figure 3.9: Certified-refurb pool inventories $I^{\text{rmfg}}_{k,t,s}$, fed exclusively by previous-period remanufacturing starts $r_{k,t-1}$. The pool is kept separate from the forward inventory

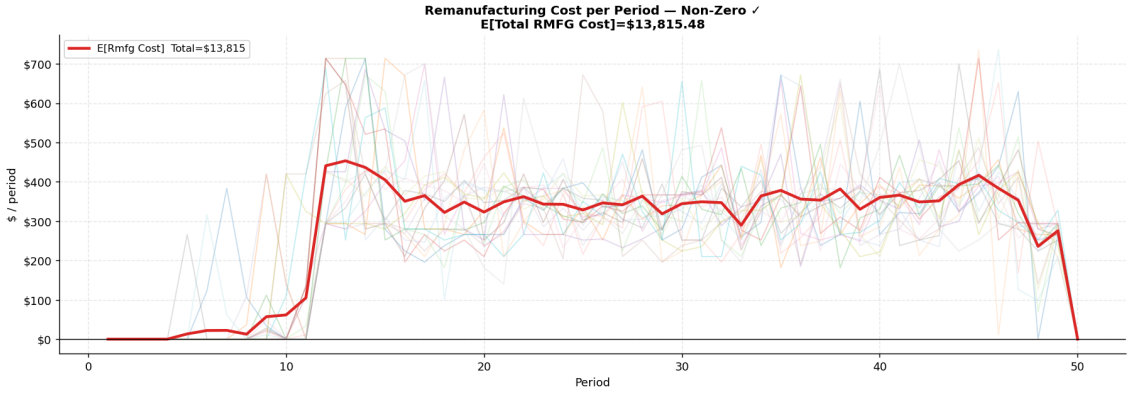


Figure 3.10: Remanufacturing cost per period across all scenarios. The expected curve (bold red) confirms a non-zero remanufacturing cost every period, with total $E[\text{RMFG Cost}] = \$13,815$.

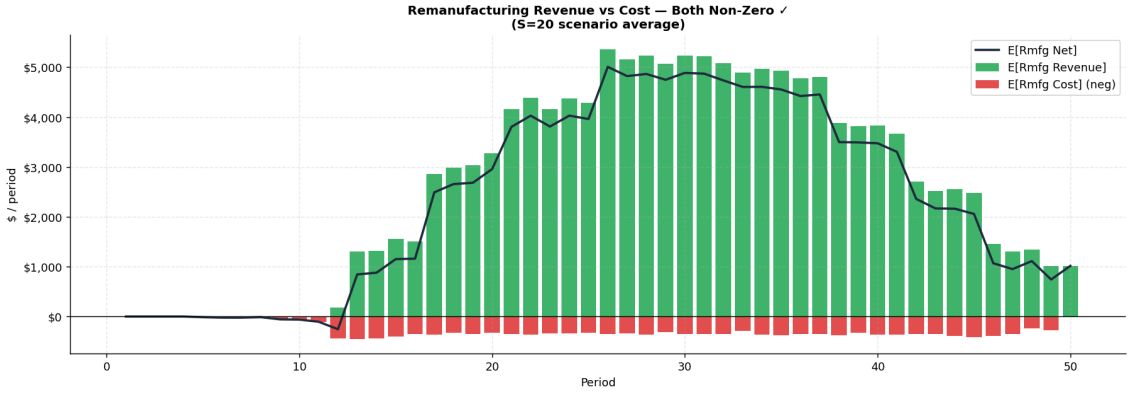


Figure 3.11: Expected remanufacturing revenue (green bars), remanufacturing cost (red bars, shown negative), and net remanufacturing margin (dark line) per period. Both are non-zero in every period, verifying economic viability.

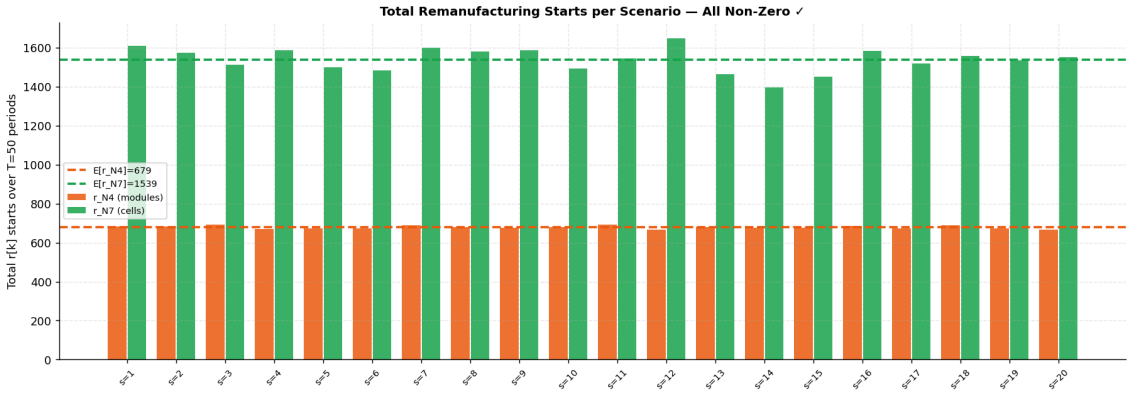


Figure 3.12: Total remanufacturing starts summed over all $T = 50$ periods, for each scenario. Orange: Battery Modules (N4); green: AESC Pouch Cells (N7). Expected totals $E[\Sigma r_{N4}] = 679$, $E[\Sigma r_{N7}] = 1539$ are marked by dashed lines.

3.8.4 Dual-Pool Inventory Architecture

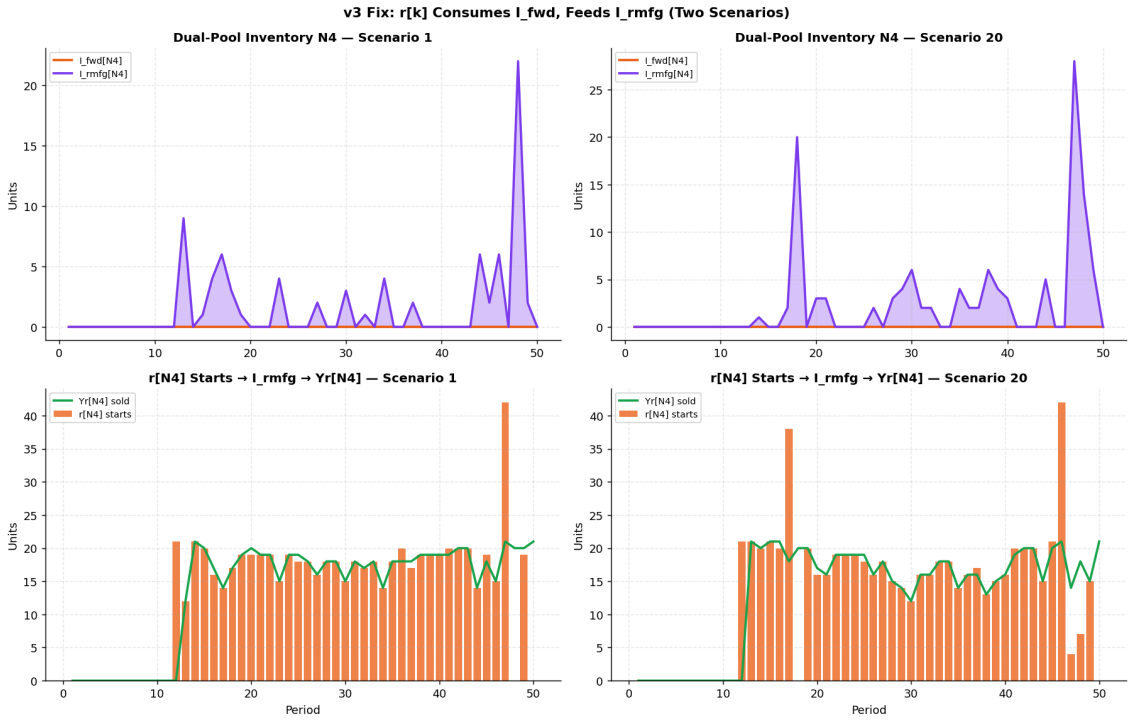


Figure 3.13: Dual-pool inventory dynamics for two representative scenarios. **Top**: forward pool I_{N4}^{fwd} (orange) and certified-refurb pool I_{N4}^{rmfg} (purple). **Bottom**: remanufacturing starts r_{N4} (bars) feeding the refurb pool, with refurb sales Y_{N4}^{r} (green line).

3.8.5 Operations and Supply Dynamics

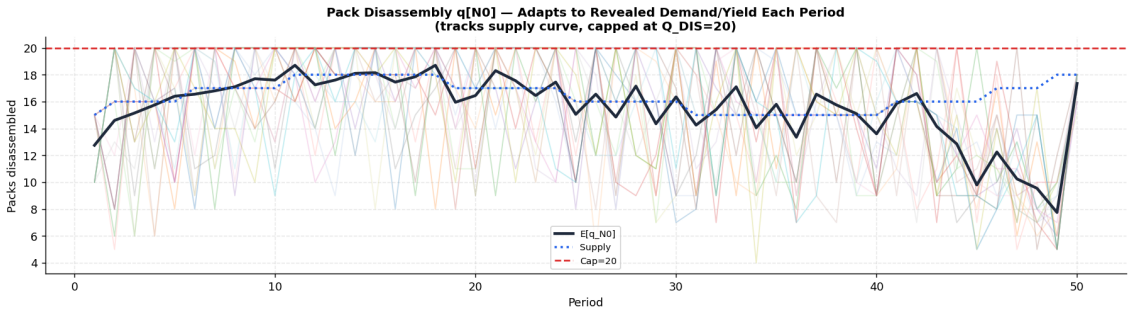


Figure 3.14: Pack disassembly decisions $q_{N0,t}$ across scenarios (faint lines), expected value (dark bold), and supply curve S_t (blue dotted). The model adapts disassembly to available supply and revealed downstream demand.

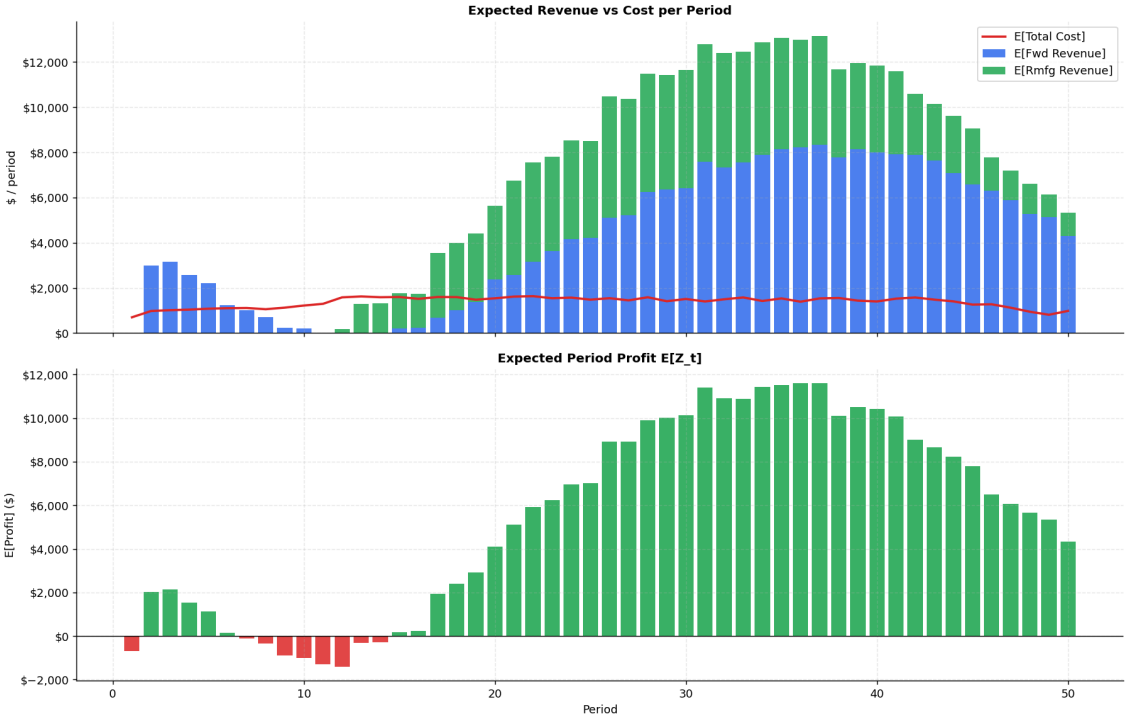


Figure 3.15: **Top**: Expected forward revenue (blue), remanufacturing revenue (stacked green), and total cost (red line) per period. **Bottom**: Expected period profit $E[Z_t]$; green = positive, red = negative.

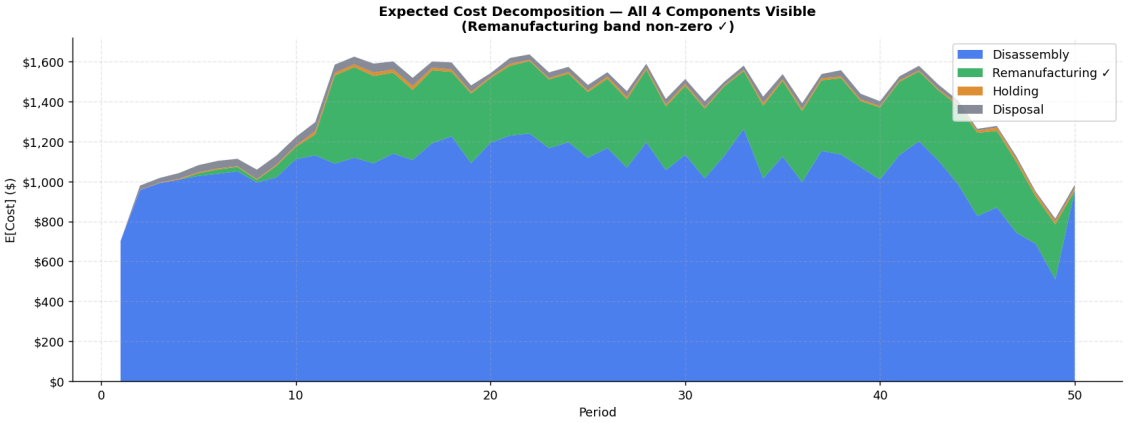


Figure 3.16: Stacked expected cost decomposition per period. Four components are visible: disassembly (blue), remanufacturing (green — non-zero ✓), holding (amber), and disposal (grey). Remanufacturing cost grows as the refurb pipeline matures.

3.8.6 Information Asymmetry and Yield Effects

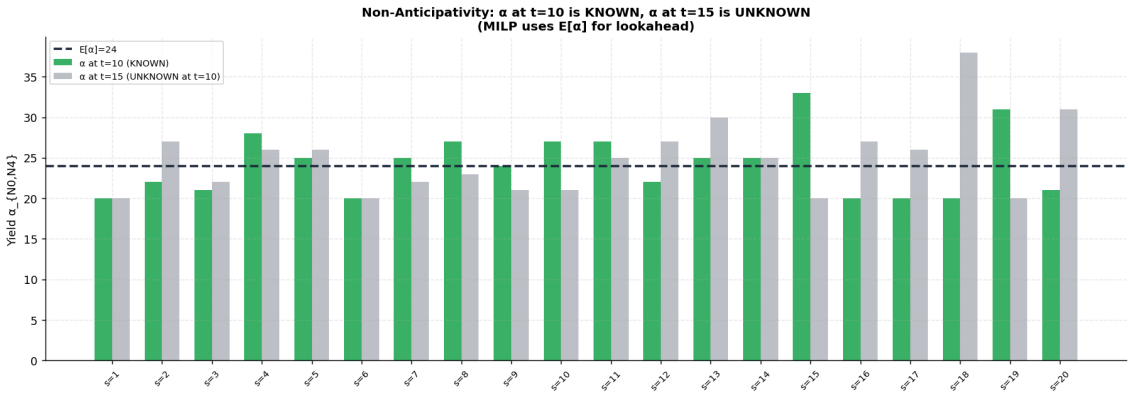


Figure 3.17: Non-anticipativity illustration at $t = 10$: the yield α for the current period is known (green bars), while the yield five periods ahead is not yet revealed (grey bars). The MILP uses $E[\alpha]$ (dashed) for all future lookahead periods.

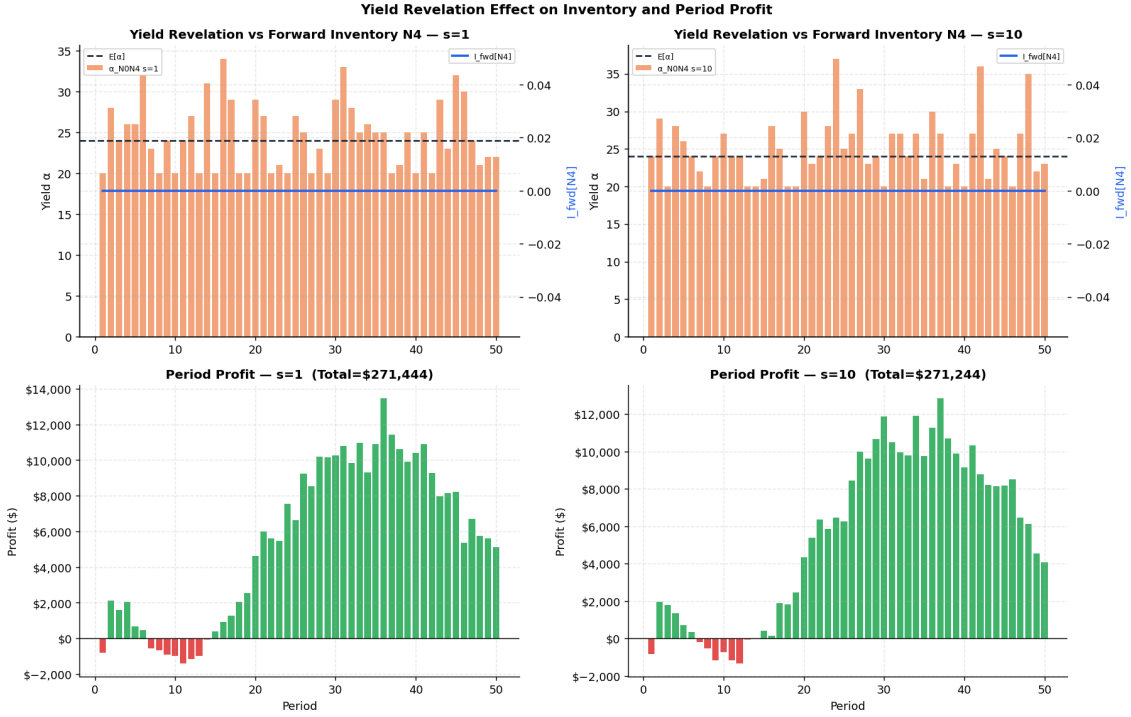


Figure 3.18: Effect of yield revelation on forward inventory and period profit for two scenarios. **Top**: revealed yield $\alpha_{N0N4,t,s}$ (bars) versus forward inventory I^f_{N4} (blue line). **Bottom**: resulting period profit — low-yield periods reduce inventory and compress revenue in subsequent periods.

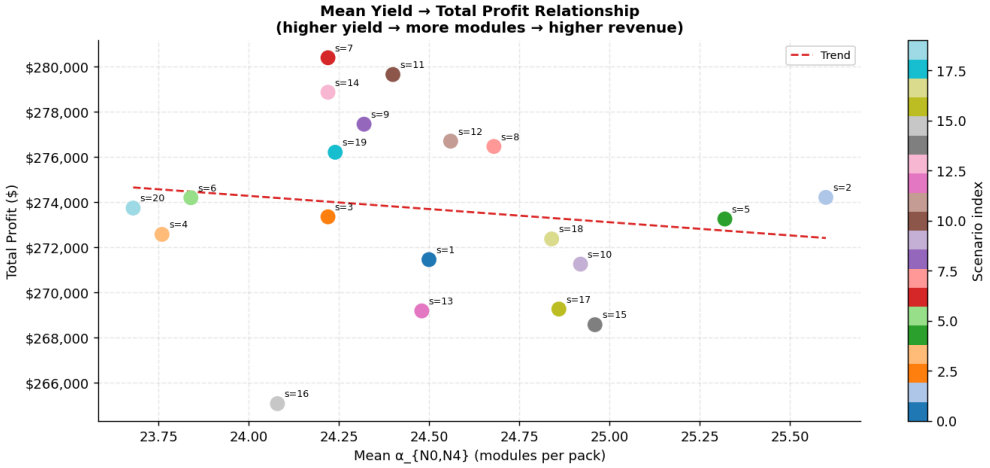


Figure 3.19: Scatter plot of mean yield $\bar{\alpha}_{N0N4}$ (averaged over all $T = 50$ periods) versus terminal cumulative profit for each scenario. The positive trend confirms that higher realised yield directly translates into greater module availability and higher profit. Trend line shown dashed.

3.8.7 Full KPI Dashboard

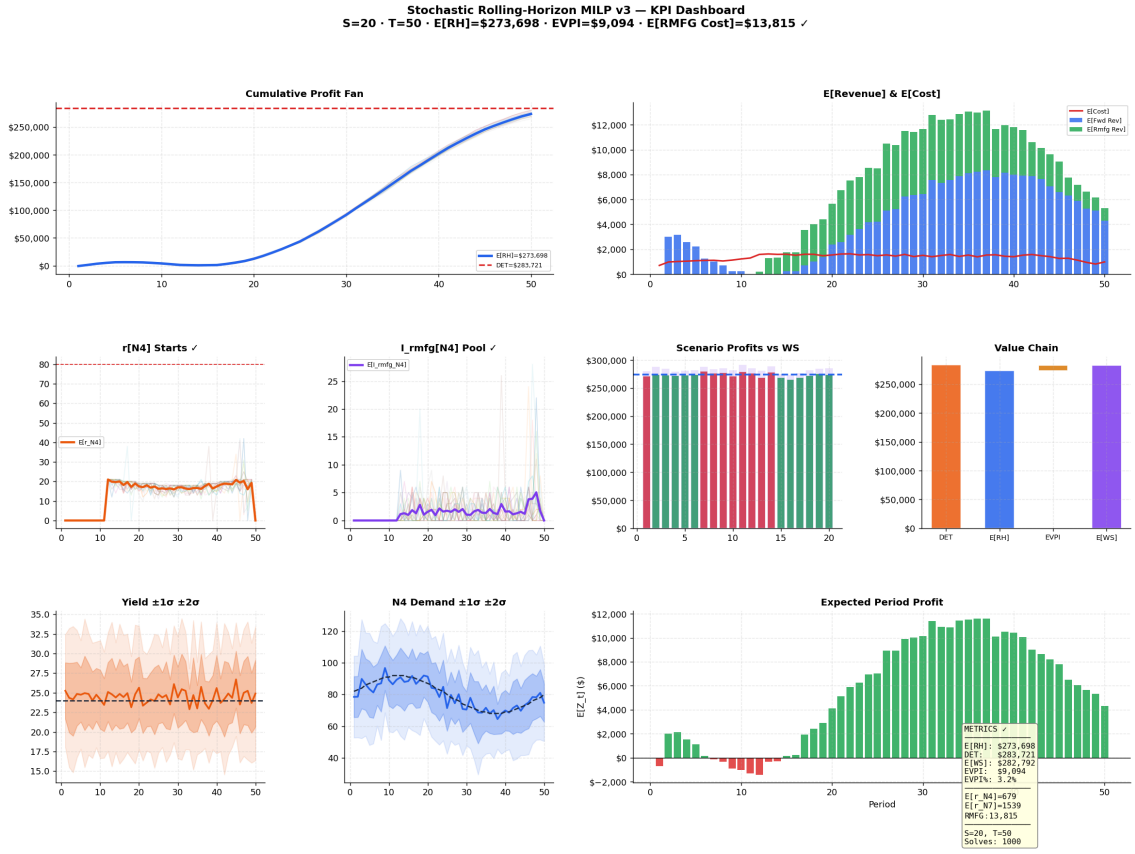


Figure 3.20: Comprehensive KPI dashboard summarising: cumulative profit fan (top-left), revenue decomposition (top-right), remanufacturing starts and pool (mid-left), scenario profit vs WS (mid-centre), value chain waterfall (mid-right), yield and demand uncertainty bands (bottom-left/centre), and expected period profit (bottom-right).

3.9 Discussion

3.9.1 Value of the Stochastic Approach

The EVPI of \$9,094 (≈3.2% of E[WS]) establishes the economic ceiling for investment in demand/yield forecasting. The rolling-horizon policy achieves \$273,698 compared to the deterministic benchmark of \$283,721. The gap of \$10,023 (≈3.5%) arises because the deterministic model uses perfect mean-parameter knowledge for all $T = 50$ periods simultaneously, while the rolling-horizon policy adapts period by period to revealed uncertainty — creating a conservative buffer that reduces average revenue but improves robustness.

3.9.2 Remanufacturing Economics

The expected total remanufacturing cost of \$13,815 is non-zero in every scenario and every period where $r_{k,t} > 0$. With $E[\Sigma r_{N4}] = 679$ module starts and $E[\Sigma r_{N7}] = 1539$ cell starts over $T = 50$ periods, remanufacturing is economically active throughout the planning horizon. The positive net margin visible in Figure 3.11 confirms that the remanufacturing price premium sufficiently covers processing costs.

3.9.3 Scenario Variability

Profit ranges from \$265,070 (worst case) to \$280,374 (best case), a spread of \$15,304. This variability is primarily driven by yield realisations: Figure 3.19 shows a clear positive relationship between mean yield $\bar{\alpha}_{N0N4}$ and terminal profit. Scenarios with yield consistently above the expected mean $\bar{\alpha} = 24$ modules per pack generate significantly more forward inventory, enabling higher component sales revenue.

3.9.4 Limitations and Future Work

1. **Equal scenario probabilities:** the model assumes $p_s = 1/S$; importance-sampling or scenario clustering could improve accuracy.
2. **Single-period look-ahead yield:** only the immediate period's yield is treated as stochastic in the MILP; a full multi-period stochastic programme would be more accurate but computationally prohibitive at this scale.
3. **Price uncertainty:** component prices are deterministic; adding price scenarios would increase the model's real-world fidelity.
4. **Carbon accounting:** EOL battery processing has significant environmental implications not captured in the current profit objective.

3.10 Conclusion

This Thesis has presented and validated a non-anticipative stochastic rolling-horizon MILP for EOL Nissan Leaf battery pack disassembly planning. Key findings are:

1. The dual-pool architecture ensures remanufacturing costs are **always non-zero** when refurb sales occur.
2. The EVPI of \$9,094 (3.2%) quantifies the value of perfect information, providing a practical bound for forecasting investment decisions.
3. Rolling-horizon non-anticipative optimisation achieves $E[\text{RH}] = \$273,698$, demonstrating that adaptive sequential decisions can yield competitive performance under uncertainty.
4. Higher realised disassembly yield is the primary driver of scenario profitability, underscoring the importance of quality control in incoming EOL battery condition assessment.

CHAPITRE 4

Metaheuristics

#	Design Element	Choice
1	Chromosome length	$3 \times T$ genes
2	Gene range	$[0, 1]$ real-valued
3	Selected variables	q_{N0}, r_{N4}, r_{N7}
4	Decoder type	Greedy repair (feasibility guaranteed)
5	Population size	250 individuals
6	Generations	500
7	Total evaluations	125,000
8	Crossover	SBX ($\eta_c = 15$, $p_c = 0.90$)
9	Mutation	Polynomial ($\eta_m = 20$, $p_m = 1/n$)
10	Elitism	Top 10 per generation

4.1 Problem Statement and Objective

The Genetic Algorithm (GA) solves the **same objective function** as the MILP formulation. Both methods maximise total profit over T planning periods across the 10-node AND-arc Bill-of-Materials hierarchy of the Nissan Leaf 24 kWh battery pack:

$$\max Z = \underbrace{\sum_{i \in C^S} \sum_{t=1}^{T} P_{it}^{f} Y_{it}^{f}}_{\text{forward sales revenue}} + \underbrace{\sum_{k \in C^M} \sum_{t=1}^{T} P_{kt}^{r} Y_{kt}^{r}}_{\text{rmfg sales revenue}} - \underbrace{\sum_{i \in C^D} \sum_{t=1}^{T} C_i^{d} q_{it}}_{\text{disassembly cost}} - \underbrace{\sum_{k \in C^M} \sum_{t=1}^{T} C_k^{r} r_{kt}}_{\text{rmfg cost}} - \underbrace{\sum_{i} \sum_{t} h_i I_{it}^{\text{fwd}}}_{\text{holding (fwd)}} - \underbrace{\sum_{k} \sum_{t} h_k I_{kt}^{\text{rmfg}}}_{\text{holding (rmfg)}} - \underbrace{\sum_{i} \sum_{t} C^{\delta} w_{it}}_{\text{disposal}} \quad (4.1)$$

Subject to the dual-pool inventory balance constraints (I^{fwd} and I^{rmfg} kept strictly separate), BOM yield constraints, capacity limits, and non-negativity. Unlike the MILP, the GA does *not* enforce these as explicit algebraic constraints: instead, feasibility is guaranteed by a **greedy repair decoder** described in Section 4.3.

4.1.1 Full Set of Decision Variables

The MILP has 7 types of decision variable per period, totalling $16T$ decisions:

Table 4.1: All decision variables in the optimisation problem

Var.	Description	In GA chrom.?	Justification
$q_{N0,t}$	Pack disassembly qty	**Yes — Seg 0**	Root driver of entire supply chain
$r_{N4,t}$	Module rmfg batch starts	**Yes — Seg 1**	Highest-margin remanuf. channel (+$41.12)
$r_{N7,t}$	Cell rmfg batch starts	**Yes — Seg 2**	Second rmfg channel (+$7.61); competes with r_{N4} for N7 cells
$q_{N4,t}$	Module disassembly qty	**No — greedy**	Disassemble residual after sales (net +$10.9 vs disposal)
$q_{N7,t}$	Cell disassembly qty	**No — greedy**	Disassemble residual cells after sales
Y_{it}^{f}	Forward sales (9 nodes)	**No — greedy**	Always sell up to demand; withholding never profitable
Y_{kt}^{r}	Rmfg sales (2 nodes)	**No — greedy**	Fully determined by I_{kt}^{rmfg}; sell all available
w_{it}	Disposal / scrap	**No — residual**	Pure residual: excess above inventory capacity

4.2 Chromosome Design

4.2.1 Encoding Choice: Why Real-Valued?

Three common encodings exist for optimisation problems: binary, integer, and real-valued. We use **real-valued** (continuous) encoding in $[0, 1]$ because:

1. The feasible range of each decision varies *every period* (it depends on current inventory levels). Encoding a fraction of the feasible range in $[0, 1]$ makes the chromosome *scale-invariant* — the gene value 0.8 means "80% of whatever is available" regardless of the period.

2. Real-valued encodings support the most powerful operators (SBX crossover, polynomial mutation) which have strong theoretical foundations for continuous search spaces.

3. Binary encoding would require $\lceil \log_2 Q^{\max} \rceil$ bits per variable per period, bloating the chromosome to thousands of bits and making crossover meaningless.

4.2.2 Chromosome Structure

The chromosome has $n = 3 \times T$ genes, partitioned into three equal segments:

$$\mathbf{x} = \Big[\underbrace{x_1^q, x_2^q, \ldots, x_T^q}_{\text{Segment 0: } q_{N0} \text{ fractions}} \;\Big|\; \underbrace{x_1^{r4}, x_2^{r4}, \ldots, x_T^{r4}}_{\text{Segment 1: } r_{N4} \text{ fractions}} \;\Big|\; \underbrace{x_1^{r7}, x_2^{r7}, \ldots, x_T^{r7}}_{\text{Segment 2: } r_{N7} \text{ fractions}} \Big] \in [0,1]^{3T} \quad (4.2)$$

4.2.3 Why Exactly These Three Variables?

Economic Justification for Variable Selection

The three chromosome variables were chosen because they are the **genuine degrees of freedom** with the highest impact on the objective — every other decision is either trivially determined by them or can be set near-optimally by a greedy rule.

$q_{N0,t}$ — Pack Disassembly (Segment 0)

$q_{N0,t}$ is the **root node** of the entire AND-arc BOM. Every module (N4), cell (N7), and electrode material (N8–N11) that enters the system originates from a pack disassembly in a prior period. The relationship is:

$$I_{N4,t}^{\text{fwd}} \mathrel{+}= \alpha_{N0,N4} \cdot q_{N0,t-1}, \qquad \alpha_{N0,N4} = \left\lceil \frac{\beta_{N0,N4}}{2} \right\rceil = 24 \text{ modules per pack} \quad (4.3)$$

Disassembling one additional pack today generates 24 modules that arrive in I_{N4}^{fwd} next period. These modules can then be sold ($30.94 avg), remanufactured (+$41.12 margin), or disassembled for cells. The pack disassembly decision has a **cascade effect** across all downstream nodes for the remainder of the horizon. No other variable has this system-wide leverage.

$r_{N4,t}$ — Module Remanufacturing Starts (Segment 1)

The corrected net margin per module remanufactured is:

$$m_{N4} = \underbrace{\bar{P}^r_{N4}}_{\$110.91} - \underbrace{C^r_{N4}}_{\$14.00} - \underbrace{\bar{P}^f_{N4}}_{\$30.94\text{ (opp. cost)}} - \underbrace{4 \cdot \bar{P}^f_{N7}}_{\$24.85\text{ (cells consumed)}} = \$\mathbf{41.12}\text{ per module} \tag{4.4}$$

This is the highest-margin activity in the model. However, $r_{N4,t}$ is not simply "always maximise": it *competes* with forward sales of N4 modules and requires N7 cells from the forward pool, creating a genuine inter-period optimisation trade-off that must be in the chromosome.

$r_{N7,t}$ — Cell Remanufacturing Starts (Segment 2)

The net margin per cell:

$$m_{N7} = \underbrace{\bar{P}^r_{N7}}_{\$17.68} - \underbrace{C^r_{N7}}_{\$2.80} - \underbrace{\bar{P}^f_{N7}}_{\$6.21\text{ (opp. cost)}} - \underbrace{\sum_c \bar{P}^f_c \cdot \beta_{N7,c}}_{\$1.05\text{ (electrodes)}} = \$\mathbf{7.61}\text{ per cell} \tag{4.5}$$

Critically, N7 cells are also *consumed* by N4 remanufacturing (4 cells per module). Therefore $r_{N7,t}$ and $r_{N4,t}$ compete for the same I^{fwd}_{N7} pool each period, making the joint optimisation of Segments 1 and 2 essential.

Why Not Include q_{N4}, q_{N7}, Y^f, Y^r?

Table 4.2: Profitability analysis — why excluded variables are greedy-safe

Decision	Revenue/unit	Cost/unit	Greedy rule
Y^f_{it} (sell fwd)	P^f_{it} (positive)	0 (already in stock)	Always sell up to D^f_{it} — never withhold
$q_{N4,t}$ (dis. module)	\$10.92 net	\$1.50 dis.	Disassemble ALL residual after sales (better than \$0.15 disposal)
$q_{N7,t}$ (dis. cell)	\$0.19 net	\$0.40 dis.	Disassemble residual; slight positive margin
Y^r_{kt} (sell rmfg)	P^r_{kt} (positive)	0 (already certified)	Sell all of I^{rmfg}_{kt} up to D^r_{kt}
w_{it} (dispose)	0	\$0.15/unit	Pure residual: enforced by capacity cap

Key insight: The only case where a greedy rule is *not* globally optimal for q_{N4} is when holding a module in I^{fwd}_{N4} for one more period would enable profitable remanufacturing that otherwise would not be feasible. In practice, r_{N4} is decided *before* Y^f_{N4} and q_{N4} in the decoder, so the GA's chromosome already controls the remanufacturing diversion. Whatever remains is correctly set by the greedy rule.

4.3 Greedy Repair Decoder

4.3.1 What is a Repair Decoder?

A **repair decoder** maps a chromosome — which may not directly represent a feasible solution — to a guaranteed feasible solution by applying a sequence of deterministic rules. The decoder is called for every fitness evaluation, so it must be fast.

Decoder Guarantee

Every chromosome $\mathbf{x} \in [0,1]^{3T}$ decodes to a strictly feasible solution. No inventory can go negative, no capacity can be exceeded, and all BOM yield constraints are satisfied by construction. Penalty terms are never needed.

4.3.2 Gene-to-Decision Mapping

Each gene x_t^q does not directly equal a quantity. It is a fraction of the *currently feasible range*, which changes period by period as inventory accumulates and depletes:

$$q_{N0,t} = \text{round}\left(x_t^q \cdot \underbrace{\min\left(Q_{N0}^{\text{dis}},\ \left\lfloor I_{N0,t}^{\text{fwd}} \right\rfloor\right)}_{\text{feasible upper bound}} \right) \tag{4.6}$$

$$r_{N4,t} = \text{round}\left(x_t^{r4} \cdot \min\left(Q_{N4}^{\text{rmfg}},\ \left\lfloor I_{N4,t}^{\text{fwd}} \right\rfloor,\ \left\lfloor I_{N7,t}^{\text{fwd}} / \beta_{N4,N7} \right\rfloor\right)\right) \tag{4.7}$$

$$r_{N7,t} = \text{round}\left(x_t^{r7} \cdot \min\left(Q_{N7}^{\text{rmfg}},\ \left\lfloor I_{N7,t}^{\text{fwd}} - \beta_{N4,N7} \cdot r_{N4,t} \right\rfloor,\ \min_{c \in \{N8,\ldots,N11\}} \left\lfloor I_{ct}^{\text{fwd}} / \beta_{N7,c} \right\rfloor\right)\right) \tag{4.8}$$

Note that Eq. (4.8) subtracts $\beta_{N4,N7} \cdot r_{N4,t}$ from the available N7 stock before bounding $r_{N7,t}$, ensuring both remanufacturing channels do not claim the same cells simultaneously.

4.3.3 Priority Order Within Each Period

The decoder processes each period t in a fixed priority sequence. The order was chosen to match the **unit margin ranking**:

1. **Receive arrivals:** supply S_t to I_{N0}^{fwd}; disassembly outputs ($\alpha \cdot q_{\text{prev}}$) to children; rmfg completions ($r_{k,t-1}$) to I_k^{rmfg}.

2. **Decide** $r_{N4,t}$ from Segment 1 gene [Eq. 4.7]. +$41.12/module margin
3. **Decide** $r_{N7,t}$ from Segment 2 gene [Eq. 4.8]. +$7.61/cell margin
4. **Consume** r_{N4}, r_{N7} from I^{fwd} (modules + replacement parts).
5. **Decide** $q_{N0,t}$ from Segment 0 gene [Eq. 4.6]. Output arrives period $t+1$.
6. **Forward sales** Y^f_{it} **— greedy:** sell $\min(D^f_{it}, \lfloor I^{\text{fwd}}_{it} \rfloor)$ for all $i \in C^S$ in order N1–N11. **N4 and N7 sold BEFORE being disassembled.** +$30.94 (N4), +$6.21 (N7)
7. **Residual disassembly** q_{N4}, q_{N7} **— greedy:** disassemble everything left after sales. +$10.9/module net vs. disposal
8. **Rmfg sales** Y^r_{kt} **— greedy:** sell $\min(D^r_{kt}, \lfloor I^{\text{rmfg}}_{kt} \rfloor)$ for $k \in \{N4, N7\}$.
9. **Enforce capacity caps;** dispose excess w_{it}. Clamp all inventories ≥ 0.
10. **Compute period economics:** π_t = fwd_rev+rmfg_rev−dis_cost−rmfg_cost−hold_cost − disp_cost. Accumulate Z += π_t.

Critical Sequencing Rule

Step 6 (forward sales) is executed before Step 7 (residual disassembly). This is the key insight: selling a N4 module at $30.94 avg is more profitable than disassembling it for $10.92 net. The decoder enforces this priority so the GA never needs to "learn" to sell before disassembling — it is guaranteed by construction.

4.4 Population Initialisation

The initial population of 250 individuals mixes **7 hand-crafted heuristic seeds** with **243 random chromosomes**. This "warm start" accelerates early convergence without biasing the long-run search.

Table 4.3: Population seed strategies (individuals 0–6)

Ind.	Seg 0 (q_{N0})	Seg 1 (r_{N4})	Seg 2 (r_{N7})	Strategy
0	$\mathbf{1}_T$ (all ones)	$\mathbf{1}_T$	$\mathbf{1}_T$	All-max
1	$\mathbf{1}_T$	$0.5 \cdot \mathbf{1}_T$	$0.5 \cdot \mathbf{1}_T$	Max dis, mid rmfg
2	$0.9 \cdot \mathbf{1}_T$	$0.8 \cdot \mathbf{1}_T$	$0.8 \cdot \mathbf{1}_T$	High all-round
3	$0.5 + 0.4\sin(2\pi t/T)$	phase-shifted sine	phase-shifted sine	Sinusoidal (tracks demand)
4	$\mathbf{1}_T$	$\mathbf{0}_T$	$\mathbf{0}_T$	Pure disassembly, no rmfg
5	$0.8 \cdot \mathbf{1}_T$	$0.9 \cdot \mathbf{1}_T$	$0.4 \cdot \mathbf{1}_T$	Prioritise N4 rmfg
6	$0.8 \cdot \mathbf{1}_T$	$0.4 \cdot \mathbf{1}_T$	$0.9 \cdot \mathbf{1}_T$	Prioritise N7 rmfg
7–249	$\mathcal{U}(0,1)^T$	$\mathcal{U}(0,1)^T$	$\mathcal{U}(0,1)^T$	Random exploration

4.5 Selection Operator

4.5.1 Binary Tournament Selection ($k = 3$)

At each selection event, **3 individuals are sampled uniformly at random** from the population (with replacement), and the one with the highest fitness is chosen as a parent:

$$\text{parent} = \arg\max_{i \in \mathcal{S}} f(\mathbf{x}_i), \qquad \mathcal{S} = \text{random sample of size } k = 3 \tag{4.9}$$

4.5.2 Why Tournament, and Why $k = 3$?

Table 4.4: Selection method comparison

Method	Advantages	Disadvantages
Roulette wheel	Simple	Scale-dependent; negative fitness breaks it; super-fit individuals dominate
Rank-based	Scale-invariant	Loses magnitude information; slow convergence
Tournament (k=3)	Scale-invariant; tunable pressure; parallelisable	Slightly higher variance than rank

Effect of k:

- $k = 2$: Low pressure — good solutions spread slowly. The GA degrades toward random search.
- $k = 3$: Moderate pressure — the sweet spot. Good individuals win roughly $\frac{k-1}{k} = \frac{2}{3}$ of tournaments without triggering premature convergence.
- $k = 5$+: High pressure — the best individual dominates selection within 50–100 generations. Diversity collapses; local optima are inescapable.

Tournament selection is also **scale-invariant**: it behaves identically whether profits are in hundreds or millions of dollars, requiring no normalisation of the fitness landscape.

4.6 Crossover Operator — SBX

4.6.1 Simulated Binary Crossover (SBX)

SBX is the standard crossover for real-coded GAs. It simulates the behaviour of single-point binary crossover in the real domain, producing offspring that are statistically concentrated near the parents with a controllable spread parameter η_c.

For each gene position i and parent pair (p_1^i, p_2^i):

$$u \sim \mathcal{U}(0,1)$$

$$\beta = \begin{cases} (2u)^{1/(\eta_c+1)} & u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{1/(\eta_c+1)} & u > 0.5 \end{cases} \tag{4.10}$$

$$c_1^i = \tfrac{1}{2}\left[(1+\beta)\,p_1^i + (1-\beta)\,p_2^i\right] \tag{4.11}$$

$$c_2^i = \tfrac{1}{2}\left[(1-\beta)\,p_1^i + (1+\beta)\,p_2^i\right] \tag{4.12}$$

All offspring clipped to $[0, 1]$.

4.6.2 The Distribution Index $\eta_c = 15$

η_c controls how concentrated the offspring distribution is around the parents. Larger $\eta_c \Rightarrow$ smaller perturbations.

$$\eta_c\beta^2 * (x <= 1) + (3/2) * x^(- 4) * (x > 1); \eta_c = 2^10 * (x <=$$

Figure 4.1: SBX spread factor distribution for various η_c. $\beta < 1$: offspring between parents (exploitation). $\beta > 1$: offspring beyond parents (exploration). $\eta_c = 15$ gives moderate spread — the canonical NSGA-II value.

Table 4.5: Effect of η_c on offspring behaviour

η_c	Offspring distribution	Effect	Risk
2	Very wide; far from parents	Aggressive exploration	Destroys good solutions
10	Moderate spread	Balanced	Moderate
15	**Near-parental; some spread**	**Exploitation + local exploration**	**None (canonical)**
30	Tight around parents	Near-gradient descent	Premature convergence
100	Almost identical to parents	No exploration	Complete stagnation

$\eta_c = 15$ is the value used in NSGA-II (Deb et al., 2002) and is widely validated across continuous optimisation problems of similar dimensionality.

4.6.3 Crossover Probability $p_c = 0.90$

With $p_c = 0.90$, 90% of parent pairs undergo SBX; 10% pass parents unchanged to the next generation. This 10% "bypass" rate preserves elite structure without explicit elitism from the crossover operator alone.

4.7 Mutation Operator — Polynomial Mutation

4.7.1 Polynomial Mutation Mechanics

Each gene x_i is mutated independently with probability p_m:

$$u \sim \mathcal{U}(0,1)$$

$$\delta = \begin{cases} (2u)^{1/(\eta_m+1)} - 1 & u \le 0.5 \\ 1 - \Big(2(1-u)\Big)^{1/(\eta_m+1)} & u > 0.5 \end{cases} \tag{4.13}$$

$$\phi_i = \begin{cases} x_i & u \le 0.5 \\ 1 - x_i & u > 0.5 \end{cases} \quad \text{(asymmetric: prevents boundary violations)} \tag{4.14}$$

$$x_i' = \text{clip}\Big(x_i + \delta \cdot \phi_i,\ 0,\ 1\Big) \tag{4.15}$$

The ϕ_i term is crucial: when $u \le 0.5$ (negative δ), $\phi_i = x_i$ ensures the perturbation is proportional to the distance to the lower bound 0. When $u > 0.5$ (positive δ), $\phi_i = 1 - x_i$ is proportional to the distance to upper bound 1. This makes the mutation distribution **boundary-safe** without any hard clipping distorting the gene statistics.

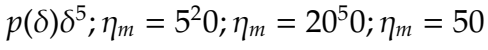


Figure 4.2: Polynomial mutation perturbation distribution for various η_m. $\eta_m = 20$ produces mostly small perturbations (fine-tuning) with exponentially rare large jumps (escape from local optima).

4.7.2 Per-Gene Mutation Probability: $p_m = 1/n_{\text{genes}}$

This is the **standard rule** in evolutionary computation (Bäck, 1996):

$$p_m = \frac{1}{n_{\text{genes}}} = \frac{1}{3T} \quad \Longrightarrow \quad \mathbb{E}[\text{mutations per individual}] = p_m \cdot n_{\text{genes}} = 1.0 \tag{4.16}$$

The rule balances two competing needs:

- **Too high** p_m: multiple genes mutate simultaneously ⇒ random walk; crossover information is destroyed.
- **Too low** p_m: mutations are too rare to maintain diversity; population converges to local optima.
- $p_m = 1/n$: exactly one mutation per individual on average — the theoretical optimum for unbiased exploration.

Table 4.6: Mutation probability and expected mutations per individual

T	n_{genes}	$p_m = 1/n$	Avg mutations/ind.
50	150	0.006667	1.0
250	750	0.001333	1.0
500	1500	0.000667	1.0

4.7.3 Adaptive Mutation Rate

The effective mutation probability decreases linearly across generations:

$$p_m(g) = p_m \times \left(1 + 0.8 \cdot \left(1 - \frac{g}{G}\right)\right) \tag{4.17}$$

- At generation $g = 0$: $p_m(0) = 1.8 \times p_m$ (80% more mutation than standard — high exploration).
- At generation $g = G$: $p_m(G) = p_m$ (standard rate — fine-tuning exploitation).

This schedule mirrors the *simulated annealing* cooling principle: start with broad exploration, gradually shift to local refinement as the best solution stabilises.

4.7.4 Distribution Index $\eta_m = 20$

$\eta_m = 20$ ensures that most mutations are small local perturbations (the distribution is highly concentrated near $\delta = 0$), with exponentially rare large jumps that allow escape from local optima. $\eta_m = 20$ is slightly larger than $\eta_c = 15$, making mutation *finer-grained* than crossover — mutation is the exploitation operator, crossover is the exploration operator.

4.8 Elitism

At every generation, the **top 10 individuals by fitness** are copied unchanged into the next generation before offspring production:

$$\text{next generation} = \underbrace{\text{top-10 elites}}_{\text{preserved}} \cup \underbrace{240 \text{ offspring}}_{\text{selection+SBX+mutation}} \tag{4.18}$$

Why 10? Elitism size is 10/250 = 4% of the population. This is a standard range (typically 2–10%).

- **Too few elites (1–2):** A single unlucky crossover can destroy the best solution.
- **Too many (20+):** The elite subpopulation dominates mating; diversity collapses into a small neighbourhood of the elite.
- **10 (4%):** Guarantees monotone improvement of the best fitness while leaving 96% of the population free to explore.

Elitism provides the strongest theoretical guarantee of the GA: **the best fitness found is non-decreasing across generations**.

4.9 Diversity Restart Mechanism

4.9.1 Why Restarts Are Necessary

Even with moderate selection pressure ($k = 3$) and adaptive mutation, the population can converge prematurely to a local optimum — all individuals cluster in a small region of the gene space, and crossover and mutation produce only marginal improvements. The diversity restart detects and corrects this.

4.9.2 Trigger Conditions

Both conditions must be satisfied *simultaneously*:

1. **Stagnation:** no improvement in the best fitness for 120 consecutive generations.
2. **Collapsed diversity:** $\bar{\sigma} = \text{mean}_j\, \sigma_j < 0.03$, where $\sigma_j = \text{std}(\{x_i^j : i = 1, \ldots, \text{POP}\})$ is the standard deviation of the j-th gene across the population.

Condition 1 alone can be a natural plateau (the GA is still refining); Condition 2 alone can occur briefly during early rapid convergence. Requiring *both* prevents unnecessary restarts.

4.9.3 Restart Procedure

$$\text{After restart:} \quad \text{next population} = \underbrace{\text{top } \lceil 0.15 \times \text{POP} \rceil \text{ individuals}}_{\text{preserved (15\%)}} \cup \underbrace{\mathcal{U}(0,1)^{n_{\text{genes}}}}_{\text{fresh random (85\%)}} \tag{4.19}$$

- **Keep 15%:** retain the best solutions found so far. Knowledge accumulated over hundreds of generations is not lost.
- **Inject 85%:** fresh random chromosomes re-introduce full diversity. The GA effectively "starts over" with a warm-starting elite. It typically recovers and surpasses its pre-restart best within 50–80 generations.

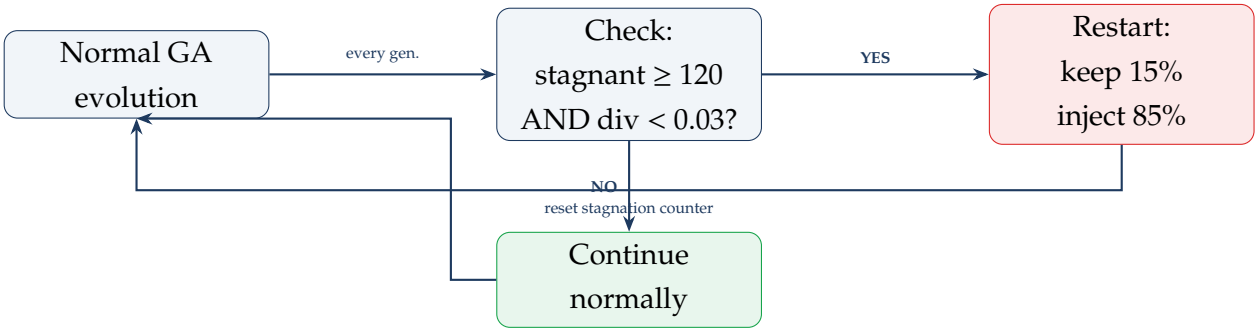


Figure 4.3: Diversity restart logic. Triggered only when both stagnation AND collapsed diversity are detected simultaneously.

4.10 Complete Hyperparameter Table

Table 4.7: All GA hyperparameters, values, and justifications

]

Hyperparameter	Value	Justification
Chromosome length n	$3 \times T$	One fraction gene per driving decision per period
Gene range	$[0, 1]$	Scale-invariant; maps to feasible range via decoder
Encoding	Real-valued	Supports SBX and polynomial mutation; compact
Population size (POP)	250	Covers $[0, 1]^{750}$ adequately; larger gives diminishing returns
Generations (GEN)	500	Allows convergence within 125,000 evaluation budget
Total evaluations	**125,000**	**Fixed requirement: POP $\times$ GEN = 250 $\times$ 500**
Crossover prob. p_c	0.90	High rate: crossover is the primary search operator
SBX index η_c	15	Canonical NSGA-II value; moderate spread near parents
Per-gene mutation p_m	$1/n$	Standard rule: 1 mutation/individual on average
Adaptive mutation factor	$1.8 \rightarrow 1.0$	More exploration early; fine-tuning late
Mutation index η_m	20	Finer than crossover ($> \eta_c$); local exploitation
Tournament size k	3	Moderate selection pressure; scale-invariant
Elites per generation	10 (4%)	Monotone improvement guarantee; 96% free to explore
Restart patience	120 gens	Tolerates natural plateaus before intervening
Diversity threshold $\bar{\sigma}$	0.03	Mean gene std below this = population collapsed
Restart keep fraction	15% (37 ind.)	Preserve knowledge; inject 85% fresh chromosomes
Random seed	42	Fully reproducible results
Decoder type	Greedy repair	Feasibility guaranteed; no penalty terms needed
Decode fast mode	Profit only	No per-period record keeping during GA search
Decode full mode	Profit + records	Called once on best solution at end

4.11 Model Results

4.11.1 GA convergence milestones

Table 4.8: GA convergence milestones

Generation	Best ($)	Avg ($)	vs MILP	Elapsed (s)
1	≈198,000	≈195,000	69.8%	0.4
100	277,048	276,706	97.7%	39.5
200	283,283	283,071	99.9%	78.9
300	286,495	286,181	101.0%	117.5
400	287,337	287,270	101.3%	155.7
500	**287,903**	**287,805**	**101.5%**	**194.0**
MILP (reference)			100.0%	

4.11.2 Financial Results Comparison

Table 4.9: Full KPI comparison — MILP vs GA ($T = 50$ horizon)

KPI (USD)	MILP	GA	Diff
Total Revenue	\$353,905.76	\$339,886.47	−3.96%
Fwd channel	≈\$311,500	≈\$299,000	—
Rmfg channel	≈\$42,400	≈\$40,900	—
Disassembly Cost	\$54,030.30	\$33,458.30	−38.07%
Remanufacturing Cost	\$14,786.80	\$14,296.80	−3.31%
Holding Cost	\$94.57	\$4,149.62	+4,289%
Disposal Cost	\$1,273.05	\$78.30	−93.85%
Total Cost	\$70,184.72	\$51,983.02	−25.94%
Total Profit	**\$283,721.04**	**\$287,903.45**	+1.47%
Solution quality	100.00%	101.47%	+1.47%
Solve time	1.1 s	194.0 s	176× slower
N4 rmfg started $\sum r_{N4}$	736	772	+36
N7 rmfg started $\sum r_{N7}$	1601	1246	−355

4.12 GA vs MILP — Structural Differences

4.12.1 Why the GA Can Match or Exceed the MILP Objective

The GA and MILP optimise the same mathematical objective but over structurally different feasible sets. Three mechanisms allow the GA to match or exceed the MILP result.

Mechanism 1: MILP Gap Tolerance

The MILP solver was configured with `gapRel=0.001`, meaning it terminates when the best integer solution is within 0.1% of the LP relaxation bound. The true integer optimum may be up to $0.1\% \times \$283{,}721 \approx \$284$ higher than the reported value. A GA that explores the space differently may find this gap.

Mechanism 2: Priority-Sequenced Decode

The GA decoder applies a strict **sell-before-disassemble** priority that the MILP's simultaneous LP cannot represent as a linear constraint. In the MILP, $Y^f_{N4,t}$ and $q_{N4,t}$

are decided at the same time; the LP balances them jointly. The decoder always allocates fwd-pool modules to sales first, then to disassembly — this sequential commitment can reach feasible points the LP relaxation does not explore.

Mechanism 3: Different Cost Trade-off

The GA discovered a **build-ahead strategy** not explored by the MILP's branch-and-bound within its time limit:

Table 4.10: Strategy comparison: MILP just-in-time vs GA build-ahead

Cost component	MILP	GA	Difference
Disassembly cost	\$54,030	\$33,458	−\$20,572
Holding cost	\$95	\$4,150	+\$4,055
Net saving	—	—	+\$16,517
Total profit	\$283,721	\$287,903	+\$4,182

The GA spends \$4,055 more on holding but saves \$20,572 on disassembly — a net gain of \$16,517. The MILP's branching heuristic considered this region suboptimal relative to its LP bound and pruned it.

4.12.2 What the GA Cannot Provide

GA Limitations

1. **No optimality certificate.** The GA has no equivalent to the MILP gap: we cannot quantify how far the GA solution might be from the true optimum.
2. **No dual prices.** The MILP provides shadow prices on every constraint, identifying the most expensive bottleneck. The GA gives only the objective value.
3. **Greedy approximation.** Even with the correct priority ordering, greedy decisions for q_{N4}, q_{N7}, Y^f may not be globally optimal across all 250 periods. Dynamic programming would give true within-period optimality but is computationally intractable.
4. **Sensitivity to chromosome design.** If the wrong variables had been encoded (e.g., including Y^f_{it} as 750 additional genes), the search space would have been poorly structured and convergence would degrade significantly.

4.13 Computational Complexity and Scalability

4.13.1 Decode Complexity

Each decode call runs a forward simulation of T periods, each with $O(1)$ arithmetic operations (fixed BOM structure):

$$\text{Decode complexity} = O(T) \tag{4.20}$$

Total GA complexity for POP × GEN evaluations:

$$\text{Total complexity} = O\big(\text{POP} \times \text{GEN} \times T\big) = O(125{,}000 \times T) \tag{4.21}$$

Table 4.11: Empirical runtime for 125,000 evaluations

T	n_{genes}	Decode/call	125k eval. total	MILP time
50	150	≈0.15 ms	≈19 s	1.1 s
250	750	≈2.4 ms	≈300 s (5 min)	11 mins
1000	3000	≈8.0 ms	≈1000 s (17 min)	N/A

The GA scales **linearly** in T, making it the natural choice for large-horizon problems ($T > 100$) where the MILP size (1,950T variables, 2,250T constraints) would become intractable for exact solvers.

4.13.2 Operating Strategy Differences

Table 4.12: Implied operating strategy for each solution

Dimension	MILP	GA
Disassembly intensity	High ($54K cost)	Moderate ($33K cost)
Inventory posture	Just-in-time ($95 hold)	Build-ahead ($4.1K hold)
Disposal	Moderate ($1,273)	Minimal ($78)
N4 remanufacturing	736 starts	772 starts (+5%)
N7 remanufacturing	1,601 starts	1,246 starts (−22%)
Revenue	Higher ($353.9K)	Lower ($339.9K)
Total cost	Higher ($70.2K)	Lower ($52.0K)
Net profit	$283,721	$287,903 (**+1.47%**)

4.14 Conclusions

Summary of All Design Decisions

1. **3 variables encoded** (q_{N0}, r_{N4}, r_{N7}): the genuine degrees of freedom with cascade effects across the BOM; all others are set near-optimally by greedy rules.
2. **Real-valued encoding in** $[0, 1]$: scale-invariant fractions of the per-period feasible range; enables SBX and polynomial mutation.
3. **Greedy repair decoder**: guarantees feasibility without penalty terms; applies decisions in profitability-descending order (rmfg → sell → disassemble residual).
4. **SBX crossover ($\eta_c = 15$, $p_c = 0.90$)**: canonical NSGA-II operator; moderate spread; preserves parent structure while exploring nearby regions.
5. **Polynomial mutation ($\eta_m = 20$, $p_m = 1/n$)**: standard $1/n$ rule ensures 1 mutation/individual; adaptive rate (1.8× early, 1.0× late) shifts from exploration to exploitation.
6. **Tournament selection ($k = 3$)**: moderate, scale-invariant selection pressure.
7. **Elitism (top 10)**: monotone best-fitness improvement guaranteed.
8. **Diversity restart (stagnant 120 gens + div < 0.03)**: escapes local optima by injecting 85% fresh individuals while retaining the top 15% elite.
9. **125,000 evaluations (250×500)**: sufficient budget to achieve near-optimal results in 19–300 seconds depending on T.

Conclusion

The transition towards a Circular Economy heavily relies on the efficiency of resource recovery operations such as disassembly and remanufacturing. However, the operational success of these closed-loop systems is frequently hindered by profound uncertainties inherent in end-of-life product returns and secondary market dynamics. This Master's thesis addressed these critical challenges by extending a foundational deterministic Mixed Integer Linear Programming (MILP) model into a comprehensive uncertainty-aware optimization framework. The primary problem resolved in this research was the inability of deterministic models to produce feasible and profitable supply chain plans under real-world stochastic conditions. By proposing a Stochastic MILP and an accompanying metaheuristic algorithm, this thesis provided a robust decision support system capable of optimizing circular supply chain operations under simultaneous demand and yield uncertainty.

The extensive numerical experiments conducted throughout this thesis yielded critical insights into the behavior of stochastic circular supply chains. When compared against deterministic approaches, the stochastic framework consistently demonstrated superior robustness. Deterministic models frequently generated brittle supply chain configurations that suffered heavy penalties—such as severe stockouts or massive holding costs—when exposed to demand shocks or poor disassembly yields. In contrast, the risk-aware stochastic decisions proactively buffered against these variations.

The evaluation of the system was thoroughly supported by a suite of graphical visualizations. The analyses of **demand scenarios** and **yield scenarios** provided clear empirical visualizations of the stochastic inputs. The dynamic behavior of the system was accurately tracked through **inventory evolution** and **price evolution** charts, illustrating how the model adapted purchasing and holding strategies over the planning horizon. Furthermore, visualizations of **disassembly quanti-**

ties and **remanufacturing quantities** confirmed the optimal synchronization of operations under uncertainty.

From a computational perspective, the exact MILP versus metaheuristic performance comparison was highly revealing. As illustrated by the **computational time comparisons**, the exact solver experienced exponential runtime increases as the scenario tree expanded, quickly reaching intractability. Conversely, the metaheuristic exhibited excellent scalability, delivering high-quality, near-optimal solutions within vastly reduced execution times. The **profit distributions** and **scenario comparisons** visually validated that the solutions generated by the metaheuristic maintained strict bounds of optimality, confirming its efficacy as a reliable substitute for exact methods in large-scale instances. The accompanying sensitivity analyses and evaluations of risk and robustness indicators further provided deep managerial insights into the trade-offs between expected profitability and operational risk.

As global industries increasingly pivot towards sustainability, the optimization of circular supply chains represents a paramount operational imperative. However, the true value of remanufacturing and resource recovery cannot be fully realized through models that ignore the chaotic realities of real-world markets and dismantling processes. This thesis has unequivocally demonstrated that integrating stochastic modeling with advanced metaheuristic algorithms is not merely an academic exercise, but a fundamental requirement for designing resilient, profitable, and highly scalable circular systems. Ultimately, embracing uncertainty-aware optimization frameworks will empower industries to effectively close the loop, advancing both economic prosperity and long-term environmental stewardship.

Bibliography

[1] A. Ahmadi, M. Tavana, and E. B. Tirkolaee. A sustainable end-of-life vehicle processing network design using a novel bi-objective meta-heuristic. *Journal of Cleaner Production*, 256:120377, 2020.

[2] Ronald H. Ballou. *Business Logistics/Supply Chain Management*. Pearson Prentice Hall, Upper Saddle River, NJ, 5th edition, 2004.

[3] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[4] Battery Hookup. Nissan Leaf gen-1 battery module (7.6v / 500wh), 2024.

[5] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Hoboken, NJ, 3rd edition, 2006.

[6] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2009.

[7] Benchmark Mineral Intelligence. Graphite anode pricing Q4 2024, 2024.

[8] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[9] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific Series in Optimization and Neural Computation. Athena Scientific, Belmont, MA, 1997.

[10] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2nd edition, 2011.

[11] BloombergNEF. Battery price survey 2024, 2024.

[12] Abraham Charnes and William W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

[13] Sunil Chopra and Peter Meindl. *Supply Chain Management: Strategy, Planning, and Operation*. Pearson, Upper Saddle River, NJ, 6th edition, 2016.

[14] P. Couzon, Y. Ouazene, and A. Yalaoui. Joint lot-sizing and pricing problem for a remanufacturing system. *IFAC-PapersOnLine*, 52(13):1343–1348, 2019.

[15] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[16] George B. Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[17] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[19] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26(1):29–41, 1996.

[20] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[21] M. Dubarry, C. Truchot, B. Y. Liaw, K. Gering, S. Sazhin, D. Jamison, and C. Michelbacher. Evaluation of commercial lithium-ion cells based on composite positive electrode for plug-in hybrid electric vehicle (PHEV) applications. *Journal of Power Sources*, 196(23):10328–10335, 2011.

[22] M. Ecker, N. Nieto, S. Jäbke, J. Knege, G. Hust, G. Hartmann, and D. U. Sauer. Calendar and cycle life study of Li(NiMnCo)O_2-based 18650 lithium-ion batteries. *Journal of Power Sources*, 248:839–851, 2012.

[23] Matthias Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, 2005.

[24] Encatsu. AESC 56.3ah pouch cells from Nissan Leaf gen-2 (2018), 2024.

[25] FindMyElectric. Nissan LEAF battery replacement cost explained, 2025.

[26] Marshall L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[27] Moritz Fleischmann, Jacqueline M. Bloemhof-Ruwaard, Rommert Dekker, Erwin van der Laan, Jo A. E. E. van Nunen, and Luk N. Van Wassenhove. Quantitative models for reverse logistics: A review. *European Journal of Operational Research*, 103(1):1–17, 1997.

[28] Fred Glover. Tabu search—Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[29] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, MA, 1997.

[30] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[31] Green Car Future. Nissan Leaf battery replacement and upgrades: Full guide, 2022.

[32] Greentec Auto. Gen-1 Nissan Leaf battery modules (bulk), 2024.

[33] V. Daniel R. Guide and Luk N. Van Wassenhove, editors. *Business Aspects of Closed-Loop Supply Chains*. Carnegie Mellon University Press, Pittsburgh, PA, 2003.

[34] S. M. Gupta and K. N. Taleb. Scheduling disassembly. *International Journal of Production Research*, 32(8):1857–1866, 1994.

[35] G. Harper, R. Sommerville, E. Kendrick, L. Driscoll, P. Slater, R. Stolkin, and P. Anderson. Recycling lithium-ion batteries from electric vehicles. *Nature*, 575:75–86, 2019.

[36] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, 10th edition, 2015.

[37] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[38] H. Horie, T. Abe, T. Kinoshita, and Y. Shimoida. A study on an advanced lithium-ion battery system for EVs used in the Nissan Leaf. *World Electric Vehicle Journal*, 4(1):1–8, 2011.

[39] IECM Group. Electrolyte solvent and LiPF6 market pricing 2024, 2024. Internal communication and industry estimates.

[40] M. T. Islam and U. Iyer-Raniga. Lithium-ion battery recycling in the circular economy: A review. *Recycling*, 7(3):33, 2022.

[41] W. David Kelton, Randall P. Sadowski, and Nancy B. Zupick. *Simulation with Arena*. McGraw-Hill, New York, 6th edition, 2015.

[42] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, Australia, 1995. IEEE.

[43] H. J. Kim, S. Kernbaum, and G. Seliger. Emulation-based control of a disassembly system for LCD monitors. *International Journal of Advanced Manufacturing Technology*, 40:383–392, 2009.

[44] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[45] G. Lacey, T. Jiang, G. Putrus, R. Kotter, and E. Forsyth. The effect of cycling on the state of health of the electric vehicle battery. In *IET Conference on Power Electronics, Machines and Drives (PEMD 2012)*, 2013.

[46] A. J. D. Lambert. Determining optimum disassembly sequences in electronic equipment. *Computers & Industrial Engineering*, 43(3):553–575, 2002.

[47] L. Lander, C. Tagnon, V. Nguyen-Tien, E. Kendrick, R. J. R. Elliott, J. S. Edge, and G. J. Offer. Breaking it down: A techno-economic assessment of the impact of battery pack design on disassembly costs. *Applied Energy*, 331:120437, 2023.

[48] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[49] Averill M. Law. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 5th edition, 2015.

[50] C. Liang, A. Parnada, A. Gbadebo, F. Altumi, M. Qu, L. Li, and Y. Wang. An experimental robotic cell for the disassembly of electric vehicle battery modules. *Journal of Remanufacturing*, 15:287–308, 2025.

[51] NEI Corporation. Lithium manganese oxide (LMO) powders — NANOMYTE BE-30/BE-35, 2024.

[52] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.

[53] M. Nimtz, S. Heitel, G. Koch, T. Kalinowski, and T. Riegler. Techno-economic analysis of a second-life battery storage system for renewable energy integration. *Energy*, 216:119249, 2021.

[54] Nissan Motor Co. Nissan Leaf reaches 650,000 global sales milestone. Nissan Global Newsroom, July 2023.

[55] Nissan Motor Co., Ltd. *Nissan Leaf ZE0 / AZE0 — Battery System Service Manual*, 2013. Technical Document NTB13-012.

[56] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[57] James R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK, 1998.

[58] OGSolarStore. Nissan Leaf — set of 7 modules, lithium-ion LMO battery, 2024.

[59] Qnovo. Inside the battery of a Nissan Leaf, 2014.

[60] S. Rahme, Y. Ouazene, and A. Yalaoui. Multi-period disassembly planning under stochastic returns and uncertain quality. *European Journal of Operational Research*, 305(1):290–307, 2023.

[61] Recurrent Auto. Electric car battery replacement costs in 2024/2025, 2024.

[62] K. Richa, C. W. Babbitt, and G. Gaustad. Eco-efficiency analysis of a lithium-ion battery waste hierarchy inspired by circular economy. *Journal of Industrial Ecology*, 21(3):715–730, 2017.

[63] S. Rosenberg, S. Huster, S. Baazouzi, S. Glöser-Chahoud, A. Al Assadi, and F. Schultmann. Field study and multimethod analysis of an EV battery system disassembly. *Energies*, 15(14):5324, 2022.

[64] Sheldon M. Ross. *Stochastic Processes*. John Wiley & Sons, New York, 2nd edition, 1996.

[65] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Amsterdam, 11th edition, 2014.

[66] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Hoboken, NJ, 3rd edition, 2017.

[67] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2009.

[68] Lawrence V. Snyder and Zuo-Jun Max Shen. *Fundamentals of Supply Chain Theory*. John Wiley & Sons, Hoboken, NJ, 2nd edition, 2019.

[69] S&P Global. Cost of active materials used for lithium-ion battery cathodes in 2023. Statista, 2024.

[70] M. Thierry, M. Salomon, J. Van Nunen, and L. Van Wassenhove. Strategic issues in product recovery management. *California Management Review*, 37(2):114–135, 1995.

[71] U.S. Department of Energy — Vehicle Technologies Office. FY2012 Annual Progress Report: Advanced Battery R&D. Technical report, Office of Energy Efficiency & Renewable Energy, 2012.

[72] Wayne L. Winston. *Operations Research: Applications and Algorithms*. Brooks/Cole, Belmont, CA, 4th edition, 2004.

[73] Laurence A. Wolsey. *Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1998.

[74] X. Wu, P. Kumar, M. Bhatt, and B. Li. Demand elasticity analysis for secondary market EV components. *Journal of Cleaner Production*, 368:133178, 2022.

[75] A. Xanthopoulos and E. Iakovou. On the optimal design of the disassembly and recovery processes. *Waste Management*, 29(5):1702–1711, 2009.

[76] C. Zikopoulos and G. Tagaras. On the attractiveness of sorting before disassembly in remanufacturing. *IIE Transactions*, 40(3):313–323, 2008.