Optimization approaches to design and manage robust and resilient manufacturing systems

Simon Thevenin

Habilitation à Diriger des Recherches

Optimization approaches to design and manage robust and resilient manufacturing systems

Defense date: November 21st 2023

Author: Simon Thevenin

Reviewers:
Jean-Philippe Gayon, ISIMA - Clermont Auvergne INP, Clermont Férand
Farouk Yalaoui, Université de Technologie de Troyes, Troyes
Walid Klibi, Kedge Business School, Bordeaux

Examiners:
Nicolas Zufferey, Geneva School of Economics and Management, Genève
Claudia Archetti, ESSEC Business School, Cergi
Nasser Mebarki, Nantes Université, Nantes

Guarant:
Alexandre Dolgui, IMT Atlantique, Nantes

Acknowledgement

I would like to express my deepest gratitude to the following individuals and groups who have played a crucial role in the completion of this manuscript:
The reviewers and the examiners: I would like to thank you for the time and effort you invested in reviewing and evaluating this work. Thank you for your insights and feedback.
Doctoral and postdoctoral students: I owe a debt of gratitude to the dedicated doctoral and postdoctoral students who have been the driving force behind the research presented in this manuscript. Your hard work, creativity, and unwavering commitment have been the backbone of this project.
Members of the MODELIS team: I would like to thank the members of the MODELIS team for their unwavering support, sharing their wealth of experience, and coming to my aid whenever I faced challenges.
My Ph.D. and postdoc advisors: A special thanks to my Ph.D. and postdoc advisors. They not only instilled in me a passion for research but also taught me the art of conducting high-quality research.
Dedication to family: Lastly, I dedicate this work to my beloved family, my wife, and my daughters. Thank you from the bottom of my heart for being part of my life.

Contents

Part I

Curriculum Vitae

Simon Thevenin
IMT Atlantique
4 rue Alfred Kastler, Nantes
simon.thevenin@imt-atlantique.fr

PROFESSIONAL EXPERIENCE

Assistant Professor 2018 -
IMT Atlantique, France

Postdoctoral researcher 2017 - 2018
HEC Montréal, Canada

Algorithm Expert 2015 - 2017
Quintiq, Netherlands

Teaching and research assistant 2010 - 2015
Geneva school of economics and management, University of Geneva

EDUCATION

PhD in business management (Operations research) 01-06-2015
University of Geneva, Switzerland
Title: "Metaheuristics for constrained production scheduling problems"
Thesis advisor: Prof. N. Zufferey

Master in databases and artificial intelligence 2009
University of Burgundy, Dijon, France

Bachelor of computer science 2008
Northumbria University, Newcastle, UK

License in computer science 2007
University of Burgundy, Dijon, France

TECHNICAL EXPERTISE AND LANGUAGES

Optimization	CPLEX, XPressMP, OPL, CPO
Programming	Python, Java/J2EE, C, C++, C#, Julia, XML, Matlab
Software	Pycharm, Netbean, Eclipse, Visual studio, Visual Code
Web server	Tomcat, Jonas, Rest API
Database	Oracle, Postgres, Mysql
Languages	French (mother tongue), English (fluent)

TEACHING EXPERIENCE

I gained teaching experience during my assistant professor position at IMT Atlantique and at the University of Geneva. During my career I have been the school supervisor for **7 Bachelor projects**, **6 Master's thesis**, and **7 Internships**, and I have taught the following courses:

Data driven optimization **2022-current**
Co-operative engineering program ($\approx$ 30 students, in French) IMT Atlantique

Operations Research **2020-current**
Co-operative engineering program ($\approx$ 30 students, in French) IMT Atlantique

Optimization under uncertainty **2019-current**
Engineering program ($\approx$ 20 students, in French) IMT Atlantique

Exact methods **2018-current**
Master ($\approx$ 15 students, in English) IMT Atlantique

Scheduling **2018-current**
Master ($\approx$ 15 students, in English) IMT Atlantique

Combinatorial optimization **2018-2022**
Master ($\approx$ 15 students, in English) IMT Atlantique

Object oriented programming **2019**
Engineering program (≈ 20 students, in French) IMT Atlantique

Java programming **2018**
Engineering program (≈ 20 students, in French) IMT Atlantique

Production management **2015**
Bachelor in Management (≈ 15 students, in French) University of Geneva

Operation management and modeling **2015**
Bachelor in Management (≈ 150 students, in French) University of Geneva

Transport optimization **2015**
Continuous education (≈ 20 students, in French) University of Geneva

Industrial management and logistics **2010 – 2014**
Master in Management (≈ 50 students, in English) University of Geneva

Inventory management **2010 – 2014**
Master in Management (≈ 50 students, in French) University of Geneva

Optimization and decision support **2010 – 2014**
Bachelor in Management (≈ 80 students, in French) University of Geneva

Production management **2010 – 2014**
Continuous education (≈ 20 students, in French) University of Geneva

ADMINISTRATIVE ACTIVITY IN TEACHING

- Responsible for the following courses at IMT Atlantique: Advanced operation Research, Optimization under uncertainties, and Optimization and decision aids. Each of these courses is a set of 45 h of teaching activities (courses, TP, TD) that constitute a coherent

teaching program. In the design phase, my role is to define the learning objectives, define the program, and write the syllabus (in discussion with the head of the Master). During the production of the course, I am responsible for finding teachers, planning the interventions, following up on the grade of the students, inputting the grade in the information system, taking part in the jury of the master, proceeding with the student evaluation of the course, and to follow up on student evaluation.

- Member of various juries (Master projects, international internship, internship, etc.) from 2018.

- Administrative tasks related to the management of the *Bachelor in Management* at the Geneva School of Economics and Management (2010-2014). As a teaching assistant at the University of Geneva, I was helping the responsible for the Bachelor in Management (about 800 students over 3 years) with various tasks such as writing various letters, advertising the Bachelor's courses to high school students at various events, producing various statistics, and organizing the graduation ceremony.

- Part of the scientific committee of the *Master in Management* at the Geneva School of Economics and Management (2015). I was part of the committee that evaluated the applicants for the Master's course.

SUPERVISED MASTERS STUDENTS

Tourandokht Karimi 2023

Title: *Sustainable assembly line design*

Co-supervisors: S.E. Hashemi

Yosra Mezghani 2022

Title: *Optimization of the design of assembly throughout their life cycle*

Co-supervisors: S.E. Hashemi, A. Dolgui

Es-Saiydy Mohammed Amine 2020

Title: *Produciton planning and scheduling with uncertain capacity and process duration*

Co-supervisors: F. Gamboa

Koktar Akash 2020

Title: *Reinforcement learning approach for MRP under uncertain lead times*
Co-supervisors: A. Dolgui

Mehrnoosh Mohammadi 2019
Title: *Multi-stage stochastic optimization for MRP under lead time uncertainties*
Co-supervisors: A. Dolgui

Phan Dung Nguyen 2019
Title: *Multi-stage stochastic optimization for contracting in ATO*
Co-supervisors: A. Dolgui, Y. Adulyasak

SUPERVISED PHD STUDENTS

Joseph Thompson 2023-2027
Title: *Machine learning approach in metaheuristics*
Co-supervisors: N. Larichi, P. Meyer, M. Mohammedi
Supervision rate: 20 %

Haed Tavakkoli-Moghaddam 2022-2025
Title: *Assembly line balancing with cobots*
Co-supervisors: A. Dolgui, O. Hazir, M. Agi
Supervision rate: 20 %

David Tremblet 2021-2024
Title: *Machine learning to improve the accuracy of mathematical model*
Co-supervisors: A. Dolgui
Supervision rate: 60 %

Dan Luo 2020-2023
Title: *Predictive and prescriptive analytics for production planning*
Co-supervisors: A. Dolgui
Supervision rate: 60 %

Metzker Paula 2019-2022
Title: *Robust optimization for lot sizing under yield uncertainty*
Co-supervisors: A. Dolgui, Y. Adulyasak
Supervision rate: 30 %

Ehsan Hashemi 2018-2021

Title: *Combinatorial optimization for the configuration of workforce and equipment in reconfigurable assembly lines*

Co-supervisors: A. Dolgui, S. Kovalev

Supervision rate: 30 %

SUPERVISED POSTDOC STUDENTS

Ilhem Slama 2020-2021

Topic: *Dissembly planning under uncertainty*

Co-supervisors: A. Dolgui, O. Ben Ammar

Ehsan Hashemi 2020-2021

Topic: *Flexible assembly line balancing*

Co-supervisors: A. Dolgui

Benham Vahdani 2020-2021

Topic: *Logistic model during a pandemic*

Co-supervisors: M. Mohammedi, P. Meyer, A. Dolgui

Erfan Asgari 2022

Topic: *Dynamic pricing in assembly to order*

Co-supervisors: M. Mohammedi

Milad Elyasi 2022-2023

Topic: *Markov decision process for flexible assembly lines*

Co-supervisors: A. Derrien

Ehsan Hashemi 2022-2023

Topic: *Prescriptive analytics for assembly lines*

Co-supervisors: A. Dolgui

RESEARCH PROJECT

ALICIA, 610,695 €, *H2020/RIA*, 2023-2026

My role: *Principal investigator from IMT Atlantique and leader of work package 2*

ALICIA aims to create and demonstrate a circular manufacturing ecosystem for production resources such as robotic arms or conveyor belts. The underlying vision is that within five to ten years, production resources will be traded and reused to their maximum utility in-between factories in Europe, ultimately contributing to "closing the loop" of production assets as circular economy subjects.

NORTON, 50,000 €, *CARNOT*, 2022

My role: *Participant*

NORTON investigates the benefit of dynamic pricing in assemble-to-order production systems, and the project aims to provide a methodology to create robust purchasing and pricing plans in this context. It is an interdisciplinary project between three domains: Machine learning, operations research, and industrial engineering.

LOGVID, 50,000 €, *CARNOT*, 2021

My role: *Participant*

This project tackles the logistics issues in a pandemic by developing a new optimization framework. A new dynamic optimization framework and an approximation solution algorithm are developed to solve the model using the demand predicted by machine learning techniques. Finally, the proposed models are used for the design and management of logistics systems.

ASSISTANT, 882,553 €, *H2020/RIA*, 2020-2023

My role: *Scientific Manager*

With a multidisciplinary consortium combining key skills in AI, manufacturing, edge computing and robotics, ASSISTANT aims to create intelligent digital twins through the joint use of machine learning, optimization, simulation, and domain models. The resulting tools will design and operate complex collaborative and reconfigurable production systems based on data collected from various sources such as IoT devices.

DREAM, 20,000 €, *GFA* , 2020

My role: *Principal Investigator*

The industrial circular economy offers an opportunity for companies to act more sustainably

and produce in a more resource-efficient manner. However, to close the cycle successfully, companies must disassemble used parts easily and cost-effectively, and this requires new methods for automated and flexible disassembly. The DREAMS project investigates the possibility of using existing assembly lines to disassemble end-of-life products for the circular economy.

OSPP, 55,000 €, *A2020 Challenge International*, 2019-2022

My role: *Principal Investigator*

The objective of this project will be to develop tools for planning/scheduling production systems considering uncertain parameters and levers available to companies to react to these uncertainties. The proposed tools will help make decisions in a reactive way, and they will allow greater flexibility for companies while minimizing their production costs..

PACE, 30,000 €, *IRT Jules Vernes*, 2019-2022

My role: *Participant*

Modernization of a production workshop based on the elements of industry 4.0 (robotization, data-based optimization, simulation, etc.).

PULSAR, 10,000 €, *Pays de la Loire*, 2019-2020

My role: *Principal Investigator*

Starting grant corresponds to a package with the creation of a community of research starting the same year as me, various seminars on research activities, mentoring, and funding of an intern that worked on production planning under uncertainty.

OSPCL, 10,000 €, *A2020 Amorçage*, 2018-2019

My role: *Principal Investigator*

A supply chain planning tool centrally optimizes sourcing, inventory, distribution, and production decisions in the supply chain. In practice, the parameters (e.g., demand, delivery times, etc.) used to plan are often uncertain. In this context, the production plan is updated in each period to consider the new information available. The objective of this project is to develop stochastic optimization models that consider the dynamic aspect of the decision process.

ADMINISTRATIVE ACTIVITIES IN RESEARCH

- Scientific Manager of the European-funded Project ASSISTANT (2020-2023). I have the responsibility of ensuring the overall scientific and technical coordination of the project, in line with the decisions of the Executive Board, including:
  - Chairing the Executive Board;
  - Ensuring the scientific and technical coordination of the ASSISTANT project, progress-monitoring of the different work packages, and following up on the decisions taken by the Executive Board;
  - Being responsible for the project achievement in line with the Grant Agreement, on time and within budget;
  - Bottom-up and top-down communication within the consortium and the Project Management Board (project management, funding, scientific and technical work), and external communication;
  - Organizing annual project meetings and initiating technical meetings within WP;
  - Undertaking all necessary legal and ethical responsibilities and obligations;
  - Ensuring maximum transparency for all partners, keeping them informed of the project progress;
  - Mediating disputes within the consortium. I enable the Consortium to avoid bottlenecks, promoting close collaboration between the strategic and operational levels to ensure fast and consistent decision making.
- Member of the scientific committee of the *19th EU/ME workshop on metaheuristics for industry* in Geneva. I took part in the selection of abstracts presented to the conference.
- Guest Editor of a special issue in the Journal of Intelligent Manufacturing.
- Member of the organizing committee for the following conferences:
  - 10th IFAC Conference Manufacturing Modelling, Management, *Nantes*, 2022
  - IFIP International Conference on Advances in Production Management Systems, *Nantes*, 2021
  - 19th EU/ME workshop on metaheuristics for industry, *Geneva*, 2018.
  - Swiss OR Days, *Geneva*, 2013.

- Organizer of special sessions:
  - Data-driven manufacturing decision, *MIM 2022*, Nantes.
  - Learning and robust decision support system for manufacturing, *APMS 2021*, Nantes
  - Workforce scheduling and line balancing at *EURO 2021*, Athens.
  - Workforce planning, *MIM 2019*, Berlin.
- Referee for the following international journals: European Journal of Operations Research, Production and Operations Management, Computers and Operations Research, Transportation Science, OMEGA, IEEE Transactions on Automation Science and Engineering, International Journal of Production Research, Theoretical Computer Science, Annals of Operations Research, Expert System With Application, Computers and Industrial Engineering, Journal of Manufacturing Systems, Journal of Intelligent Manufacturing, Advances in Mechanical Engineering, and PLOS One.
- Referee for the following conferences: AAAI 2022, MIM 2022, APMS 2021, CASE 2020, MEIE 2020, MIM 2019, and APMS 2019.

The rest of my CV provides references to my paper. I have underlined the name of my PhD students in these publications.

PEER-REVIEWED JOURNAL PAPERS

23. S. E. Hasemi-Petroodi, S. Thevenin, S. Kovalev, A. Dolgui, "Markov decision process for multi-manned mixed-model assembly lines with walking workers", *International Journal of Production Economics,* **255**, 108661 (2023). DOI:10.1016/j.ijpe.2022.108661

22. P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "Robust optimization for lot-sizing problems under yield uncertainty", *Computers and Operations Research,* **149**, 149 (2023). DOI:10.1016/j.cor.2022.106025

21. D. Tremblet, S. Thevenin, P. Meyer, A. Dolgui, "Makespan estimation in a flexible job-shop scheduling environment using machine learning", *International Journal of Production Research*, (2023). DOI:10.1080/00207543.2023.2245918

20. B. Vahdani, M. Mohammadi, S. Thevenin, P. Meyer, A. Dolgui, "Production-sharing of critical resources with dynamic demand under pandemic situation: The COVID-19 pandemic", *Omega*, (2023). DOI:10.1016/j.omega.2023.102909

19. B. Vahdani, M. Mohammadi, S. Thevenin, M. Gendreau, A. Dolgui, P. Meyer, "Fair-split distribution of multi-dose vaccines with prioritized age groups and dynamic demand: The case study of COVID-19", *European Journal of Operational Research,* **310**, 1249-1272 (2023). DOI:10.1016/j.ejor.2023.03.032

18. G. Castañé, A. Dolgui, N. Kousi, B. Meyers, S. Thevenin, E. Vyhmeister, P-O. Östberg, "The ASSISTANT project: AI for high-level decisions in manufacturing", *International Journal of Production Research,* , (2023). DOI:10.1080/00207543.2022.2069525

17. E Vyhmeister, G Castañé, P.-O. Östberg, S. Thevenin, "A responsible AI framework: pipeline contextualisation", *AI and Ethics,* **3**, 175-197 (2023). DOI:10.1007/s43681-022-00154-8

16. D. Luo, S. Thevenin, A. Dolgui, "A state-of-the-art on production planning in Industry 4.0", *International Journal of Production Research,* **61**, 6602-6632 (2022). DOI:10.1080/00207543.2022.2122622

15. S. Thevenin, Y. Adulyasak, J.-F. Cordeau, "Stochastic dual dynamic programming for multi-echelon lot-sizing with component substitution", *Inform Journal on Computing,* **34**, 3151-3169 (2022). DOI:10.1287/ijoc.2022.1215

14. I. Slama, O. Ben-Ammar, S. Thevenin, A. Dolgui, "Stochastic program for disassembly lot-sizing under uncertain component refurbishing lead times", *European Journal of Operations Research,* **303**, 1183-1198 (2022). DOI:10.1016/j.ejor.2022.03.025

13. S. Thevenin, N. Brahimi, O. Ben-Ammar, "Robust optimization approaches for purchase planning with supplier selection under lead time uncertainty", *European Journal of Operations Research,* **303**, 1199-1215 (2022). DOI:https://doi.org/10.1016/j.ejor.2022.03.029

12. S. E. Hasemi-Petroodi, S. Thevenin, S. Kovalev, A. Dolgui, "Model dependent task assignment in multi-manned mixed-model assembly lines with moving workers", *Omega,* **113**, 102688 (2022). DOI:10.1016/j.omega.2022.102688

11. C. Gonnermann, S. E. Hashemi-Petroodi, A. Dolgui, S. Thevenin, A. Dolgui, R. Daub, "A skill-and feature-based approach to planning process monitoring in assembly planning.", *The International Journal of Advanced Manufacturing Technology,* **122**, 2645-2670 (2022). DOI:10.1007/s00170-022-09931-5

10. S. Thevenin, N. Mebarki, P. Chatellier, "Dynamic scheduling of a picking robot with limited buffer and rejection: an industrial case study", *International Transaction in Operations Research,* **29**, 1394–1416 (2021). DOI:10.1111/itor.13050

9. S. Thevenin, Y. Adulyasak, J.-F. Cordeau, "Stochastic optimization for material requirements planning", *Production and Operations Management,* **30**, 475–493 (2021). DOI:10.1111/poms.13277

8. S.E. Hashemi Petroodi, A. Dolgui, S. Kovalev, M. Y. Kovalyov, and S. Thevenin, "Workforce planning in manufacturing systems as a factor of their reconfigurability: a state of the art", *International Journal of Production Research*, 1–24 (2020). DOI:10.1080/00207543.2020.1823028

7. S. E. Hashemi-Petroodi, S. Thevenin, S. Kovalev, A. Dolgui, "A survey on operations management methods for hybrid human-robot collaborative systems in manufacturing", *Annual Reviews in Control,* **49**, 264–276 (2020). DOI:10.1016/j.arcontrol.2020.04.009

6. S. Thevenin, N. Zufferey, "Learning variable neighborhood search for a scheduling problem with time windows and rejections", *Discrete Applied Mathematics,* **261**, 344–353 (2019). DOI:10.1016/j.dam.2018.03.019

5. S. Thevenin, N. Zufferey, R. Glardon, "Model and metaheuristics for a scheduling problem integrating procurement, sale and distribution decisions", *Annals of Operations Research* **259**, 437–460 (2018). DOI:10.1007/s10479-017-2498-z

4. S. Thevenin, N. Zufferey, J.-Y. Potvin, "Makespan minimization for a parallel machine scheduling problem with preemption and job incompatibility", *International Journal of Production Research,* **55(6)**, 1588–1606 (2017). DOI:10.1080/00207543.2016.1181285

3. S. Thevenin, N. Zufferey, M. Widmer, "Order acceptance and scheduling with earliness and tardiness penalties", *Journal of Heuristics,* **22(6)**, 849–890 (2016). DOI:10.1007/s10732-016-9321-x

2. S. Thevenin, N. Zufferey, J.-Y. Potvin, "Graph multi-coloring for a job scheduling application", *Discrete Applied Mathematics,* **234**, 218–235 (2016). DOI:10.1016/j.dam.2016.05.023

1. S. Thevenin, N. Zufferey, M. Widmer, "Metaheuristics for a scheduling problem with rejection and tardiness penalties", *Journal of Scheduling,* **18(1)**, 89–105 (2015). DOI:10.1007/s10951-014-0395-8

JOURNAL PAPERS CURRENTLY UNDER REVIEW

4. Y. Mezghani, S. E. Hashemi Petroodi, S. Thevenin, A. Dolgui, "Design and reconfiguration planning of mixed-model assembly lines under uncertain evolutions of product family", *under review in the International Journal of Production Research*, (2023).

3. I. Slama, O. Ben-Ammar, S. Thevenin, A. Dolgui, "A stochastic model for a multi-echelon disassembly lot-sizing problem under random lead times", *under review in Transportation Research Part E*, (2023).

2. S. E. Hashemi Petroodi , S. Thevenin, S. Kovalev, A. Dolgui, "An adversarial approach for the robust balancing of mixed-model assembly lines with dynamic task assignment", *under review in European Journal of Operations Research*, (2023).

1. P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "Adjustable robust optimization for lot-sizing problems under yield uncertainty", *under review in European Journal of Operations Research*, (2023).

PUBLICATIONS IN CONFERENCE PROCEEDINGS

21. D. Tremblet, S. Thevenin, A. Dolgui, "Predicting makespan in flexible job shop scheduling problem using machine learning", *MIM 2022*, (June 2022). DOI:10.1016/j.ifacol.2022.09.305

20. S. E. Hashemi-Petroodi, S. Thevenin, A. Dolgui, "Mixed-model assembly line design with new product variants in production generations", *MIM 2022, IFAC papers online* **55 (10)**, 25-30 (June 2022). DOI:10.1016/j.ifacol.2022.09.363

19. M. Paul, S. Thevenin, J. Schulz, N. Brahimi, H.H. Benderbal, A. Dolgui, "Advancing circular economy: Research roadmap for circular integrated production systems.", *ProVE 2021*, (November 2021). DOI:10.1007/978-3-030-85969-5_75

18. D. Lamy, S. Thevenin, "The group shop scheduling problem with power requirements", *PMS2021*, (April 2021).

17. P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "The adaptive robust lot-sizing problem with backlogging under demand uncertainty", *CASE 2021*, (August 2021). DOI:10.1109/CASE49439.2021.9551425

16. P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "Optimization for lot-sizing problems under uncertainty: a data-driven perspective", *APMS 2021*, (September 2021). DOI:10.1007/978-3-030-85902-2_75

15. D. Luo, S. Thevenin, A. Dolgui, "A digital twin-driven methodology for material resource planning under uncertainties", *APMS 2021*, (September 2021). DOI:10.1007/978-3-030-85874-2_34

14. M. Mohammadi, S. Thevenin, A. Dolgui, "Multi-period Multi-item Lot sizing under Random Yield and Stochastic Capacity", *MIM*, (November 2020).

13. S.E. Hashemi Petroodi, S. Thevenin, S. Kovalev, A. Dolgui, "The impact of dynamic tasks assignment in a paced mixed-model assembly line with moving workers", *APMS*, 509–517 (September 2020). DOI: 10.1007/978-3-030-57997-5_59

12. S. Thevenin, O. Ben-Ammar, N. Brahimi, "A robust approach for the joint lot-sizing and supplier selection", *Proceedings of the 21th Annual Congress of the French Operations Research Society*, (February 2020).

11. S.E., Hashemi Petroodi, C. Gonnermann, M. Paul, S. Thevenin, A. Dolgui, R. Gunther, "Decision support system for joint product design and reconfiguration of production systems", *APMS*, (September 2019). DOI: 110.1007/978-3-030-30000-5_30

10. C. Selma, S. Thevenin, N. Mebarki, O. Cardin, D. Thiériot, L. Bruggeman, "Heuristics for robots-humans tasks assignment in a containers loading center", *IMS*, (August 2019). DOI:10.1016/j.ifacol.2019.10.011

9. A. Dolgui, S.E. Hashemi Petroodi, S. Kovalev, M.Y. Kovalyov, S. Thevenin, "Workforce planning and assignment in mixed-model assembly lines as a factor of line reconfigurability: state of the art", *MIM*, (August 2019). DOI:10.1016/j.ifacol.2019.11.623

8. S. Thevenin, N. Zufferey, "Enhancing variable neighborhood Search with a learning mechanism", *Proceedings of the 20th Annual Congress of the French Operations Research Society*, (February 2019).

7. S. Thevenin, N. Zufferey, "Reactive variable neighborhood search", *The 19th International Conference on Metaheuristics for Industry*, (Mars 2018).

6. S. Thevenin, N. Zufferey, "A deconstruction-reconstruction metaheuristic for a job scheduling problem", *The 5th international conference on metaheuristics and nature-inspired computing*, (October 2014).

5. S. Thevenin, N. Zufferey, "Variable neighborhood search for a scheduling problem with time window penalties", *14th International Conference on Project Management and Scheduling PMS*, 226–229 (April 2014).

4. S. Thevenin, N. Zufferey, J.-Y. Potvin, "Multi-objectives parallel machines scheduling with incompatible jobs", *Proceedings of the 15th Annual Congress of the French Operations Research Society***(selected for the final phase of the young researcher prize)**, (February, 2014).

3. S. Thevenin, N. Zufferey, M. Widmer, "Tabu search for a single machine scheduling problem with discretely controllable release dates", *Proceedings of the 12th International Symposium on Operational Research (SOR)*, 103 – 108 (September 2013).

2. S. Thevenin, N. Zufferey, J.-Y. Potvin, "Tabu search for a preemptive scheduling problem with job incompatibilities", *Manufacturing, Modeling, Management, and Control (MIM)*, 1590–1595 (June 2013). DOI:10.3182/20130619-3-RU-3018.00049

1. S. Thevenin, N. Zufferey, M. Widmer, "Tabu search for a single machine scheduling problem with rejected jobs, setups and deadlines", *9th International Conference on Modeling, Optimization & SIMulation MOSIM*, (June 2012).

BOOK CHAPTER

- O. Battaïa, L. Benyoucef, X. Delorme, A. Dolgui, S. Thevenin, "Sustainable and energy efficient reconfigurable manufacturing system", *Chapter in Advanced Manufacturing (Springer, 2019), Eds. L. Benyoucef.*
- N. Zufferey, J. Respen, S. Thevenin, "All-terrain tabu search approaches for production management problems", *Chapter in Recent Development of Metaheuristics (Springer, 2018), Eds. L. Amodeo, E-G. Talbi, F. Yalaoui.*

WORK IN PROGRESS

- P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "Distributionally robust optimization for lot-sizing under yield uncertainty".
- M. Elyasi, S. Thevenin, A. Cerqueus, A. Dolgui, "Mixed-model assembly lines with process time uncertainty".
- D. Tremblet, S. Thevenin, A. Dolgui, "Data-driven models for the integrated lot-sizing and flexible job-shop scheduling problem".
- D. Luo, S. Thevenin, A. Dolgui, "A fix and optimize approaches for lot-sizing under dynamic type demand uncertainty".
- S. Thevenin, Y. Adulyasak, " Scalable multi-item inventory planning for slow-Movers: A mathematical programming approach".
- E. Asgari, M. Mohammadi, S. Thevenin, "Inventory management in assemble to order system with returns".
- P. Metzker, S. Thevenin, Y. Adulyasak, A. Dolgui, "Optimization under uncertainties for the lot-sizing problem: a state of the art".
- D. Nguyen, S. Thevenin, Y. Adulyasak, A. Dolgui, "Multi-stage stochastic optimization for contracting in assemble to order systems".

INVITED SEMINARS

9. Hyper-dynamic workers in the digitalized factory, *Rennes*, CUTIS seminar at Rennes School of Business September, 2022

8. Keynote presentation on learning and robust decision support system for manufacturing decisions, *Nantes*, MIM 2022 June 24, 2022

7. Presentation of the project ASSISTANT to the AI pathway workshop of the connected factory (EFFRA), *Brussels*, June 15, 2022

6. Data-based optimization for supply chain planning, *PASCA (Online)*, May 2, 2020

5. Material requirement planning under stochastic demand , *GERAD (Montréal, Canada)*, May 2, 2018

4. Stochastic optimization for supply chain planning, *IMT Atlantique (Nantes, France)*, April 5, 2018

3. Modeling concurrent production planning and scheduling, *Procter & Gamble (Geneva, Switzerland)*, April 15, 2015

2. Balancing assembly lines, *Ecole Polytechnique Fédéral de Lausanne*, February 17, 2015

1. Production planning and distribution, *Ecole Polytechnique Fédéral de Lausanne*, May 12, 2015

LARGE AUDIENCE PUBLICATIONS

3. "Artificial Intelligence in manufacturing", *White paper of the AIM-NET network*, (2023).

2. "AI as an enabler for long-term resilience in manufacturing", *White paper of the Word Manufacturing Foundation*, (2021).

1. "Le hasard fait mieux les choses", *Atlanstic 2020 - Magazine*, (2020).

CONFERENCE PRESENTATIONS

17. Robust lot-sizing and supplier selection under lead time uncertainty, *EURO 2022, Helsinki, Finland*, July 2022

16. A robust approach for the joint lot-sizing and supplier selection, *Seminaire SLP 2020, Nantes, France*, February 2020

15. Stochastic dual dynamic programming for multi-echelon lot-sizing, *Seminaire group OAD 2020, Nantes, France*, July 2019

14. Stochastic dual dynamic programming for supply chain manufacturing master planning, *POMS International Conference 2019, Brighton, UK*, September 2019

13. Stochastic dual dynamic programming for multi-echelon lot-sizing, *IMT Workshop on Optimization, Nantes, France*, July 2019

12. Stochastic optimization for supply chain planning under demand uncertainty, *9eme Colloque IMT systèmes de production du futur, Paris, France*, Octobre 2018

11. Multi-echelon multi-product capacitated lot-sizing problem with lead time and uncertain demand, *EURO 2018, Valencia, Spain*, July 2018

10. Scenario based stochastic optimization for the multi-echelon lot-sizing problem, *International Symposium on Mathematical Programming 2018, Bordeaux, France*, July 2018

9. Stochastic optimization formulations for the multi-echelon multi-item capacitated lot-sizing problem, *Optimization Days 2018, Montréal, Canada*, May 2018

8. Metaheuristics for integrated production scheduling, express material deliveries, and outbound distribution, *Optimization Days 2017, Montréal, Canada*, May 2017

7. A multi-coloring problem with constrained color class sizes, *Optimization Days 2014, Montréal, Canada*, May 2014

6. Mutual exclusion scheduling with preemption, *Spring Seminar in Operations Research, Zinal, Switzerland*, January 2014

5. Hybrid meta-heuristics for an order acceptance and scheduling problem with discretely controllable release dates, *11th Swiss OR Days, Geneva, Switzerland*, May 2013

4. Heuristics for a k-multi-coloring problem with rejections, *Spring Seminar in Operations Research, Zinal, Switzerland*, January 2013

3. A tabu search algorithm for a single machine scheduling problem with earliness and tardiness penalties, *International Colloquium on Graphs and Optimization (GO). Leukerbad, Switzerland*, August 2012

2. Evolutionary techniques for an order acceptance problem, *3rd Student Conference on Operational Research (SCOR), Nottingham, UK*, April 2012

1. Heuristics for a single machine scheduling problem with rejection and setup costs, *Spring Seminar in Operations Research, Zinal, Switzerland*, January 2012

Part II

Scientific contributions and research perspectives

Chapter 1

Introduction

This document presents the research I conducted after my PhD on optimization approaches for the lot-sizing problem under uncertainty and for the assembly line design problem. All works reported in this document have been published in international journals, or the related articles are currently under review. Therefore, I have only summarized here the most important scientific contribution I made on different topics, and I provide pointers to the journal articles for readers interested in more details on a specific topic, or additional methods. This document does not report detailed numerical results, but the analyses of extensive computational experiments are available in the journal articles. The rest of this introduction describes my career path, an overview of the research topic I worked on during the last years, the notations used in the manuscript, and the organization of this report.

1.1 Career path

After completing my Master's in database and artificial intelligence, I started my PhD on metaheuristics applied to scheduling problems under the supervision of Nicolas Zufferey. After my PhD, I joined Quintiq, a software provider for various planning puzzles, including production planning and production scheduling. My job involved developing a generic production scheduling solver that can solve the large majority of scheduling problems faced by the clients of the firm. After almost two years of work, we delivered this optimizer and tested it on one of the most challenging scheduling puzzles encountered by the firm. This experience in the industry helped me realize I wanted to pursue an academic career.

During my PhD, I discovered the field of operations research. While the focus of my PhD was on metaheuristics, I learned about exact methods in PhD schools, conferences, and

discussions with the members of the operations research community. I had the feeling that learning about these approaches would strengthen my knowledge. Therefore, after working for two years at Quintiq, I restarted my academic career, with a postdoc on Benders decomposition method. I contacted Jean-François Cordeau, who introduced me to Yossiri Adulyasak. They obtained a grant for a two-year postdoc on stochastic programming for supply chain planning. During these two years, I learned about mathematical programming and Benders decomposition, and optimization under uncertainty. A part of the funds for the postdoc came from IVADO, an institute on artificial intelligence in Montreal. This picked my interest in other topics in AI during this postdoc, such as in machine learning or reinforcement learning.

After almost two years of postdoc, I joined IMT Atlantique and the research team MODELIS. I work closely with Alexandre Dolgui, who shared his experience in managing PhD students and postdoctoral researchers and applying for and conducting research projects. Since 2018, I significantly extended the work on lot sizing under uncertainty. I also started to work on the assembly line balancing problem to integrate within the team MODELIS. I applied my knowledge on solving lot sizing under uncertainty to the assembly line design problem. These two optimization problems have different structures. While the assembly line design problem is purely combinatorial, lot sizing involves both continuous and binary variables. Thus, the two problems present different challenges, and they require different solution methods.

Throughout my career, I had the chance to work on the three decision levels of a manufacturing system, namely, process planning (strategical), production planning, and scheduling. While this document focuses on the two research topics I explore after my PhD (assembly line design, and tactical planning under uncertainty), I have not completely abandoned my research on metaheuristics for scheduling problems. (Thevenin and Zufferey, 2019) and (Thevenin et al., 2022c) are two examples of my contributions to scheduling problems made after my PhD.

1.2 Research Projects

Different research projects have helped to fund the works described in this document, and I hired PhD students and postdocs with these funds. I started working on assembly line balancing when Alexandre involved me in the supervision of the PhD thesis of Ehsan Hashemi. This thesis took place in the RMS project funded by the Région Pays de la Loire. One year later, we launched a collaboration with HEC Montreal on lot sizing under uncertainty.

Following my postdoc at HEC Montréal, we built the project OSPP which was co-funded by Atlanstic2020 and the research chair of Yossiri. During this project, I co-supervised the thesis of Paula Metzker with Yossiri Adulyasak and Alexandre Dolgui.

The ASSISTANT project funded a large proportion of the research presented in this document. ASSISTANT is a research and innovation action funded by Horizon 2020. Its coordinator is IMT Atlantique, and I am its scientific manager. The project involves 12 partners (academics or industrials) and has a budget of around 6M euros. I built the project around the three decision levels (process planning, production planning, and scheduling) that lead to the main combinatorial optimization problem encountered in a manufacturing facility. The project started in 2020, and I conducted most of my recent research activities over the past year in this project. The project funds the PhD thesis of Dan Luo and David Tremblet, as well as the postdoc of Ehsan Hashemi and Milad Elyasis. Dan Luo is currently working on stochastic lot sizing. David Tremblet explores the research area on translating machine learning models to learn parts of the constraints of a mathematical model. Milad Elyasis is working on Markov Decision Processes to model uncertain process durations in assembly line design. Ehsan Hashemi is working on robust optimization for process design, and he is investigating the links between simulation models and optimization for process design.

1.3 Research topic

Nowadays, the manufacturing industry operates in a much more complex and volatile environment than in the past decades. The increase in the size of the production mix is in part responsible for the increase in the complexity of the manufacturing system and the supply chain. To remain competitive, firms tend to offer a large assortment of products to their customers. For instance, we commonly find workshops assembling hundreds of different model variants. This leads to challenges in designing assembly lines that can efficiently handle such a large family of model variants. The production of a large assortment of items also leads to complex inventory management problems because a manufacturer may have to manage thousands of components. In addition, the product assortment change frequently, and this creates instability and uncertainties in the manufacturing system. Companies tend to shorten product life cycles to provide their customers with the latest technologies before their competitors. Modification of the product family every 6-12 months is a common trend. Thus, companies modify the assembly lines and supply network frequently to follow the changes in

production requirements. Moreover, the large number of end items and the frequent changes in product lead to uncertainties in operations management. It is more difficult to accurately predict the demand for each individual item in a large collection of items than for a single item. The lack of regularity in the production process prevents manufacturers from controlling the production system, which leads to quality issues and machine breakdowns. Dealing with a large number of suppliers that supply a large number of components also increases the chances of delay in the arrival of the required components and materials. Besides, recent crises have shown that global supply chains are not always reliable.

Manufacturing processes are becoming complex and various parameters cannot be predicted accurately. To overcome this challenge, manufacturers require advanced software tools that can design and manage manufacturing systems and supply chains to make them robust, resilient, flexible, and reconfigurable. In this context, my focus is on two key problems: assembly line design and lot sizing.

In assembly line design, we assign tasks, workers, and equipment to stations. The objective is to design a line that balances the load among stations and minimizes the cost of the line. We aim to design assembly lines that can easily adapt to changes in demand. On the one hand, we consider the flexible assignment of workers and tasks to adapt to the variation of the production load in each takt during production. On the other hand, we study reconfiguring the line to meet the changing production requirements. A reconfiguration requires stopping the line for several days (or a week) to move the equipment and to qualify the line. In contrast, the time to move workers in a flexible line is negligible.

Lot-sizing problems are encountered in several software (supply chain planning, material requirement planning, among others). The lot-sizing problem (LSP) decides the amount to order/produce/distribute in each period and for each node of the network. We aim to provide plans that are robust against various sources of uncertainties, such as customer demand, supplier delivery lead times, and others.

My recent works enhance the LSP and assembly line design models to make robust and resilient decisions. Robustness is the ability to perform well under various conditions, whereas resilience is the ability to adapt and recover. To provide robust decisions, we enhance the assembly line design and lot-sizing model to account for various uncertainties. To make the suggested solution resilient, we incorporate dynamic decision making. The resulting optimization model accounts for the recourse action available for each scenario and suggests solutions where the system adapts efficiently to a wide range of situations.

To provide tools that can cope with this highly uncertain environment, we rely on new

technologies available on the manufacturing shop floor. Industry 4.0 fosters the digitalization of shop floors, and we can collect an increasing amount of data from shop floors and from various software. Exploiting this data can help us build probabilistic forecasts or uncertainty sets that act as input for optimization under uncertainty approaches. New technologies also include smart manufacturing equipment (e.g., robots, cobots, CNC machines, etc.) that can change their behavior with simple software-level changes. These types of equipment bring flexibility and help in adapting the line when the product requirement changes.

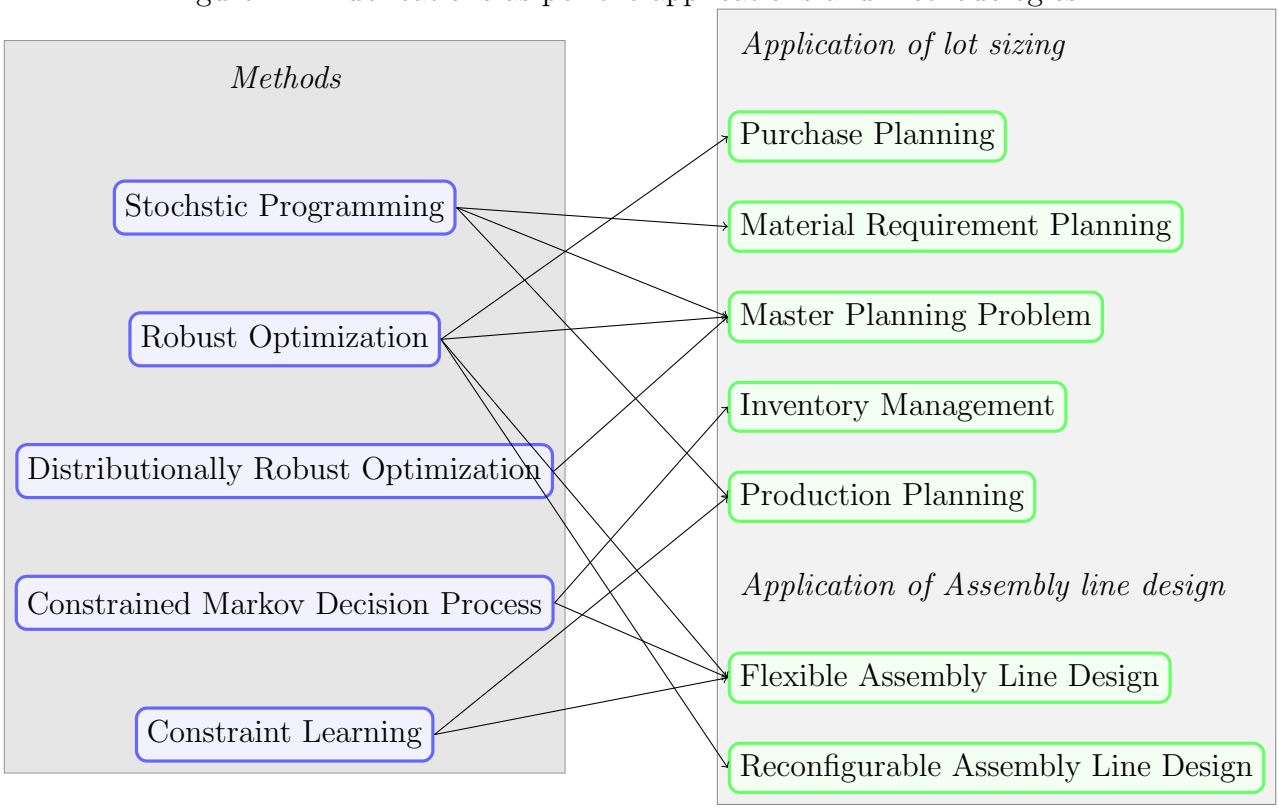
I work with three families of optimization under uncertainty approaches, namely, stochastic programming (SP), robust optimization (RO), and solution of constrained Markov decision process (MDP). Stochastic programming requires providing a probability distribution for unknown parameters. One model minimizes the expected objective function, whereas others (called chance constraint) ensure that the probability to meet a constraint is above a given threshold. Robust optimization searches for the solution that minimizes the worst-case cost or the solution that satisfies a constraint in the worst case. Robust optimization approaches are more conservative and require defining the uncertainty set properly to include only realistic scenarios. Further, robust optimization approaches are less demanding in terms of computational time than approaches that solve stochastic programs. I was also recently interested in distributionally robust optimization approaches. Distributionally robust optimization applies the robust perspective on stochastic optimization because it minimizes the expected cost against the worst-case distribution in a well-defined ambiguity set. Finally, the constrained Markov decision process applies to dynamic decision processes that respect the Markov property, and they are often employed in inventory management.

Incorporating the imprecision of some parameters enriches mathematical models, and the resulting model produces more accurate results than relying solely on deterministic versions. We explored a novel approach to further enrich these models by utilizing machine learning to learn certain constraints automatically. In (Luo et al., 2022), we highlight the similarities between the resulting model and the digital twin. The definition of a digital twin in the literature does not specify the type of model used for the physical or virtual twin, and we propose the creation of digital twins where the virtual model is a mathematical program. To be characterized as a digital twin, the model must accurately represent the physical twin and consider uncertainty when parameters cannot be estimated accurately. Constraint learning helps represent the physical twin accurately and change the model automatically to follow changes on the shop floor. A major challenge in the realization of a digital twin is connecting the model to the physical object (the manufacturing system). Solving this challenge requires

expertise in real-time data acquisition (which I do not possess), and on the integration of data from different sources. In the recent multidisciplinary research project ASSISTANT (Castañé et al., 2022), we collaborated with experts in data acquisition and data modeling to address these issues. We present some promising results in (Gonnermann et al., 2022).

Figure 1.1 shows the various optimization approaches I have considered over the past few years and links these methods to the considered software application. Table 1.1 classifies the papers I have published (or submitted) to international journals according to the application and methodology. Four out of the six works on metaheuristics for scheduling were performed during my PhD thesis, and none of the six are reported here.

Figure 1.1: Publications as per the applications and methodologies



1.4 Notations

For the sake of readability, Tables 1.2 and 1.3 provide the notations used in Section 2 and 3, respectively. Each chapter considers a different optimization problem, and each chapter uses its own set of notations. Therefore, while I kept some consistency between the notation

Table 1.1: Publications as per applications and methodologies

	Scheduling	Lot sizing	Assembly line design
Determinstic	Thevenin et al. (2015, 2016, 2017a,b, 2018); Thevenin and Zufferey (2019); Thevenin et al. (2022c)		
Stochastic programming		Slama et al. (2022); Thevenin et al. (2021, 2022a)	
Robust optimization		Metzker et al. (2023a,b); Thevenin et al. (2022b)	Hashemi-Petroodi et al. (2024, 2022)
Distributionally robust optimization		Metzker et al. (2024)	
Constrained/coupled MDPs		Thevenin and Adulyasak (2024)	Hashemi-Petroodi et al. (2023); Elyasi et al. (2024)
Constraint learning		Tremblet et al. (2024a,b)	Gonnermann et al. (2022)
State-of-the-art		Luo et al. (2022)	Hashemi-Petroodi et al. (2020, 2021)

among the different chapters, the notations in each chapter are defined independently. We denote vectors and matrix with boldface; if A is a set, we denote by $\mathbf{X_{Ab}}$ the vector with elements X_{ab} for all $a \in A$. When a decision variable X in a model becomes a parameter of another model, its value is denoted by $\breve{X}$. When the context is clear, we drop the index of the parameter or variable. For instance, when discussing multi-item lot sizing, we denote the demand of item i in period t in scenario ω by d_{it}^{ω}. However, in the context of the deterministic single item, the demand is denoted by d_t. We identify random parameters with a tilde. For instance, $\tilde{d}_{it}$ denotes the random demand of item i in period t. When a parameter takes value in a finite set, we use a hat, reversed hat, and bar to denote its maximum, minimum, and average value, respectively. For instance, $\hat{d}_i t$, $\check{d}_{it}$, $\bar{d}_{it}$ denote the minimum, maximum, and average value of the demand of item i in period t. When a heuristic requires recording the best-found solution, the values of the variable in the best-found solution are denoted with a star. For instance, $X_{it}^{\star}$ denotes the quantity of item i to order in period t in the best-found solution.

1.5 Organization of this document

This HDR thesis contains 2 parts. Part I provides my academic resume, and part II is a manuscript that describes my main scientific contribution. Part II is divided into 5 chapters.

Symbol	Meaning
i, j	Index of items
k	Index of resources
p	Index of inventory level
q	Index of quantity
t, τ	Index of periods
ω, ω', ξ	Index of scenarios
$\mathcal{I}$	Set of items
$\mathcal{I}_e$	Set of end items (with customer demand)
$\mathcal{I}_c$	Set of components (with internal demand only)
$\mathcal{K}$	Set of resources
$\mathcal{P}$	Set of inventory position
$Q(p)$	Set of quantity that can be ordered when the inventory level is p
$\mathcal{T} = \{1, \ldots, T\}$	Set of periods
Ω	Set of scenarios
Ξ	Set of scenarios sampled from Ω
Φ_{it}	Set of scenarios for item i in period t in the reformulation
b_i	Unit backorder cost for item i
C_{kt}^{ω}	Capacity of resource k in period t in scenario ω
d_{it}^{ω}	Demand for item i in period t in scenario ω
e_i	Setup cost for item i
h_i	Unit holding cost for item i
K_{ik}^{ω}	Unit resource consumption for item i on resource k in scenario ω
$\check{L}$	Minimum value of the lead time
$\hat{L}$	Maximum value of the lead time
o_k	Unit extra capacity cost for resource k
r_{ij}	Number of item in node i required to produce one unit in node j
r'_{ij}	Number of item j obtained by disassembling item i
R^k	Capacity of resource k
v_{it}	Unit cost for item i in period t
Γ	Budget of uncertainty
ϕ_{pq}^{ki}	Consumption of resource k when item i has stock level p and we order q units
$\rho_{it\tau}^{\omega}$	Proportion of item i ordered in period t received in period τ in scenario ω
B_{it}^{ω}	Backlog level of item i in period t in scenario ω
f_t	Optimal costs in periods $t+1$ to T given the decision made in stage t
H_{it}	Worst-case inventory and backlog costs for item i in period t
I_{it}^{ω}	Inventory level of item i in period t in scenario ω
O_{kt}^{ω}	Extra capacity of resource k in period t in scenario ω
S_{it}	Target stock level for item i in period t in a base stock policy
s_{it}	Re-order for item i in period t in a base stock policy
V_{pq}^{i}	Probability to order quantity q when the stock level for item i is p
X_{it}^{ω}	Order quantity for item i in period t in scenario ω
Y_{it}^{ω}	Equal 1 if an order of item i is passed period t in scenario ω, and 0 otherwise
W_{ijt}^{ω}	Amount of component i consumed to produce item j in period t in scenario ω
Z_t	Deviation of the uncertain parameter from its nominal value
α_{it}^{0}, $\alpha_{it}^{i\tau\tau'}$, $\beta_{it}^{j\tau}$	Weights in the affine policy

Table 1.2: Notations used in section 2.

Symbol	Meaning
i	Index of a model variant.
o, o'	Index of a task.
$\mathcal{A}$	Set of precedence relations (o, o')
$\mathcal{B}_d$	Set of action available in state d
$\mathcal{D}$	Set of states
$\mathcal{E}$	The set of equipment
$\mathcal{I} = \{1 \dots I\}$	Set of items
$\mathcal{O}$	Set of all tasks
$\mathcal{O}_i$	Set of tasks for item i
$\mathcal{S} = \{1 \dots S\}$	Set of sequentially located stations
Ω	Set of all possible sequences
C	Takt time.
c_e	Cost of equipment e
$f(Y_{11}, \dots, Y_{IS})$	Number of workers on line when Y_{is} workers are in station s for model i
$i_s^{t\omega}$	The model $i_s^{t\omega}$ processed at station s in takt t
$\hat{l}$	Maximum the number of workers assigned to the same station
p_{io}^l	Processing time of task o performed on a model i with l workers
q_s^a	The number of workers in station s for action a
R_{oe}	Equal to 1 if task o requires equipment e, and 0 otherwise
s_{it}^{ω}	Station at which the model i is processed in each takt t
u_i	Maximum number of item i in a picture of the line
α	Cost of a worker
y_{soi}^a	Equal to 1 if action a requires to process task o of model i in station s
$b_{sl}^{\omega t}$	Equals 1 if l workers are in station s in takt t of sequence ω (0 otherwise)
V_{ad}	The probability to perform actions a in state d
w_{se}	Equal to 1 if equipment e is chosen for station s, and 0 otherwise
x_{soi}	Equals 1 if task o for model i is performed at station s, and 0 otherwise
Y	Number of workers to hire
Y_{is}	Number of workers in station s when it processes item i
Z_{ad}	Equals 1 if actions a if the probability to perform action a is greater than 0

Table 1.3: Notations used in section 3.

Chapters 2 and 3 present my contributions to optimization under uncertainty in the lot-sizing and in assembly line design problems, respectively. Chapter 4 presents the intelligent digital twin concept and the works on constraint learning. Besides a summary of my contributions, Chapters 2, 3, and 4 present perspectives for future works in each field. The document ends with a short conclusion in chapter 5.

Chapter 2

Lot sizing under uncertainties

2.1 Introduction

Production planning refers to the tactical decisions required to ensure production. During this activity, the planner considers a time horizon of 6 to 24 months, and the planner decides, for each period, the quantities to order from suppliers, produce in the factories, and transport between the nodes of the supply chain. In decision support software, these tactical decisions are divided into different modules, such as material requirement planning, capacity planning, and distribution requirement planning. This division is consistent with how humans divide the problem into simpler ones, and with the managerial processes implemented in the company. In all these modules, (semi)automatic decision making is based on the solutions of the variants of the lot-sizing model.

Over the last few years, I considered the application of stochastic programming, robust optimization, and distributionally robust optimization approaches for LSPs. For each of my contributions in lot sizing under uncertainty, Table 2.1 presents the software application, uncertain parameter, and optimization under the uncertainty paradigm.

Table 2.1: Classification of my works on lot sizing under uncertainty according to the application, uncertain parameter, and type of approach, where SP, RO, and DRO stand for stochastic programming, robust optimization, and distributionally robust optimization, respectively.

Paper	Application	Uncertain parameter	Approach
Thevenin et al. (2021)	Material requirement planning	Demand	SP
Thevenin et al. (2022a)	Requirement planning with component substitution	Demand	SP
Thevenin et al. (2022b)	Purchase planning	Supplier's delivery lead time	RO
Slama et al. (2022, 2024)	Disassembly planning	Refurbishing lead time	SP
Metzker et al. (2023a,b)	Single item lot sizing	Yield rate	RO
Metzker et al. (2024)	Master scheduling	Yield rate	DRO
Luo et al. (2024)	Supply chain planning	Demand	SP

Companies have faced the challenge of managing inventory in a context where demand, lead times, and supply are uncertain. Therefore, the topic of dealing with these uncertainties in inventory management is far from new. Common approaches rely on the computation of a variety of parameters (safety stock, planned lead time, ...) that are input to the planning approach. We refer the reader to (Silver et al., 1998) for more information on inventory management. Over the last few decades, more and more works have considered the application of stochastic programming and robust optimization for LSPs. The resulting approach may be seen as the integration of safety parameters computations and lot-sizing decisions. For instance, Brandimarte (2006) compare models with the dynamic and static types of uncertainty for the multi-item capacitated lot-sizing. Tempelmeier (2013) presents different chance constraint approach that corresponds to different service level. The topic of optimization under uncertainty for lot-sizing has gained momentum over the last few years. My works were among the first to consider stochastic programming for lot-sizing in the multi-echelon bill of material under uncertain demand. We also consider the application of lot-sizing in remanufacturing facilities under refurbishing lead time uncertainty, and this is a trendy topic that deals with circular supply chains.

(Bertsimas and Thiele, 2006) and (Bienstock and Özbay, 2008) are two fundamental works on robust optimization for LSPs. Bertsimas and Thiele (2006) provided the approximation of the model that can be handled by dualization per constraint approach, and Bienstock and Özbay (2008) discuss the adversarial approach. As for stochastic programming, the use of robust optimization in lot-sizing is gaining interest quickly. While most works (Alem et al., 2018; Agra et al., 2018) consider the case of demand uncertainty, we were among the first to consider robust optimization approaches to deal with the uncertainty on yield and lead time. Our results show that uncertainties in these data result in different challenges, and more research is required for these problems.

The rest of this chapter is organized as follows. Section 2.2 defines the deterministic LSP and discusses its applications in different software. The following sections successively provide my main contributions in the stochastic LSPs (section 2.3), robust optimization for LSPs (section 2.3), and constrained Markov decision process for inventory management (section 2.5).

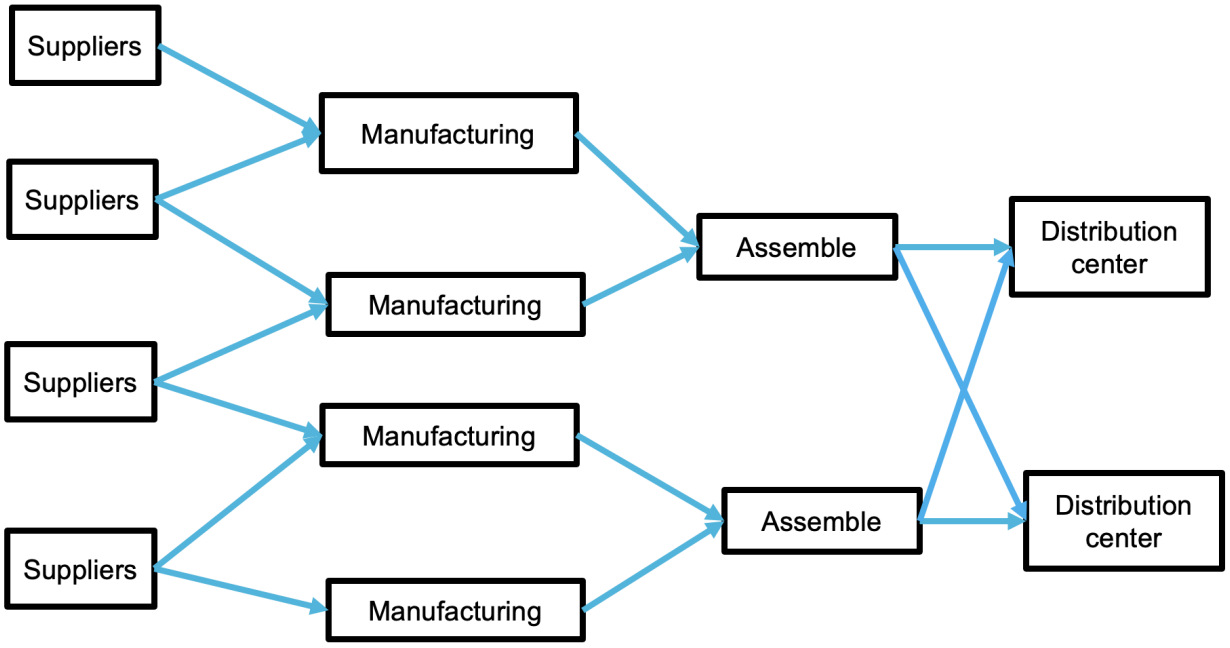


Figure 2.1: Example of a supply chain

2.2 Deterministic lot sizing

2.2.1 Formal problem definition

This section provides the reader with sufficient information about the LSP to understand the optimization problem and gain an understanding of its practical applications. I do not aim to list all possible extensions of the LSP or to provide a state-of-the-art. For more information on the variant of lot-sizing models, we refer the interested readers to (Jans and Degraeve, 2008; Voss and Woodruff, 2006). This section presents the multi-echelon capacitated lot-sizing problem (MCLSP), and the next section provides the different extensions I have studied in the past few years. The MCLSP is a generic variant of the LSP that can be applied in various software modules, such as material requirement planning and supply chain planning among others. Some modules may require simplification, such as the master schedule, where the network reduces to a single item. Other modules may require modification or extension, such as the full truckload constraints that may be required by some supply chain planning applications.

The structure of the network in which the material flows is an important input for the MCLSP. Figure 2.1 shows an example of such a network in the context of supply chain

planning. For simplicity, we call each node of the network an item, even though two nodes may represent the same item in different locations. We denote the set of items by $\mathcal{I}$. In the mathematical model, the edges of the network are represented with parameters r_{ij}, which indicate the number of items i required to produce one unit of item j. The interpretation of this network can vary depending on the application. In material requirement planning, the network represents a bill of materials, which describes the amount r_{ij} of component i required to assemble j. In manufacturing, the network is a bill of processes that gives the different processing steps required to transform raw material into a finished product, and r_{ij} equals 1 if material i is the input of the process that produces item j, and 0 otherwise. In distribution planning, the network has an out-tree shape, and r_{ij} equals 1 if shipping points i can deliver j and 0 otherwise. In supply chain planning, the supply network (shown in Figure 2.1) integrates all these components and provides the amount of material that needs to be shipped from node i to enable the production of one unit in node j. From a mathematical perspective, the concept of the network remains the same across applications, although the shape of the underlying graph may differ. A bill of processes has a chain structure, whereas a bill of materials is similar to a tree. In contrast, a supply chain network has a more general shape and can be considerably deep. In the context of disassembly, where the company recovers components from end-of-life items, the BOM has a different meaning. In this case, we consider the number r'_{ij} of component j obtained by disassembling item j. This latter case requires a slightly different formulation (see Slama et al., 2022).

The input parameters of the MCLSP include the demand d_{it} for each end-item i in period t within the planning horizon $\mathcal{T}$, as well as the cost parameters. The objective of the MCLSP is to determine the production quantity X_{it} for each item i in each period t to minimize the setup costs e_i and inventory costs h_i for all items i. To compute these different costs, the model includes variables that represent the inventory level I_{it} for item i in period t. To account for the fixed cost, the model considers an additional binary variable Y_{it} that is equal to 1 if X_{it} is positive and 0 otherwise. The practical application of the MCLSP often requires accounting for backorders, lead times, resource capacities, and production yields. I describe these elements below.

Backorder costs penalize late deliveries to customers. In deterministic contexts, late deliveries are preferred when they reduce costs or ensure a feasible plan. However, in an uncertain context, the backorder becomes important since it module the late deliveries resulting from the wrong prediction of some parameters. The MCLSP does not include backorders for components to avoid solutions where items are produced when their components are not

available. Therefore, we divide the set of items $\mathcal{I}$ into two subsets: the set of end-items $\mathcal{I}_e$ with customer demand and the set of components $\mathcal{I}_c$, which have only internal demand. A backorder cost b_i is associated with each component i in $\mathcal{I}_c$, and the model computes the backlog level B_{it} for item i in period t

The lot-sizing model can account for the available capacity in the production node. In such a case, the input includes a set of resource $\mathcal{K}$, the capacity consumption K_{ik} on resource k per unit of item i, and the available capacity C_{kt} of resource k in period t. Often, companies can increase their capacity at an extra cost, and we denote the extra capacity cost by o_{kt} for resource k in period t. To compute these costs, the model includes variable O_{kt} that represents the amount of extra capacity on resource k in period t. When an order is released in a node, it is available for the subsequent node after the lead time L_i. The lead time models transport duration between two nodes of a supply chain. The lead time may be strictly positive between two nodes that correspond to different processes in the same factory. In this case, the lead time is a flexibility provided to the scheduler. At the scheduling level, the orders may be scheduled at any time within the next L_i periods, and this flexibility may help reduce sequence-dependent setup costs for instance. In some applications, the release of a production lot outputs only a portion of these items called yield rate. The yield rate and lead time can be combined in a single parameter $\rho_{it\tau}$ that indicates the proportion of item i ordered in period t received in period τ.

The mixed integer linear programming formulation of the MCLSP is as follows:

$$\min \quad \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}(h_i I_{it} + e_i Y_{it} + v_i X_{it}) + \sum_{i\in\mathcal{I}_e}\sum_{t=1}^{t=T-1} b_i B_{it} + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} o_{kt} O_{kt} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \rho_{it\tau} X_{i\tau} + I_{i0} - \sum_{\tau=1}^{t} d_{i\tau} - I_{it} + B_{it} = 0 \qquad i\in\mathcal{I}_e, \quad t\in\mathcal{T} \tag{2.2}$$

$$\sum_{\tau=1}^{t} \rho_{it\tau} X_{i\tau} + I_{i0} - \sum_{\tau=1}^{t} \left(\sum_{j\in\mathcal{I}} r_{ij}\cdot X_{j\tau}\right) - I_{it} = 0 \qquad i\in\mathcal{I}_c, \quad t\in\mathcal{T} \tag{2.3}$$

$$X_{it} \leq M_i Y_{it} \qquad i\in\mathcal{I}, \quad t\in\mathcal{T} \tag{2.4}$$

$$\sum_{i\in\mathcal{I}} K_{ik} X_{it} \leq C_{kt} \qquad t\in\mathcal{T}, \quad k\in\mathcal{K} \tag{2.5}$$

$$B_{it} \geq 0 \qquad i\in\mathcal{I}_e, \quad t\in\mathcal{T} \tag{2.6}$$

$$I_{it} \geq 0, \quad X_{it} \geq 0, \text{ and } Y_{it} \in \{0,1\} \qquad i\in\mathcal{I}, \quad t\in\mathcal{T} \tag{2.7}$$

$$O_{kt} \geq 0 \qquad k\in\mathcal{K}, \quad t\in\mathcal{T}. \tag{2.8}$$

The objective function (2.1) minimizes the sum of the setup costs, variable costs, holding costs, backorder costs for end items, and extra capacity costs. Constraints (2.2) and (2.3) model the flow of items through the network and compute the inventory and backlog level for end items and components, respectively. Constraints (2.4) set the setup variable to 1 for all periods with production, and the capacity constraints are given in (2.5).

2.2.2 Applications of lot sizing

Model (2.1)-(2.8) is sufficiently generic to cover applications in master production scheduling, material requirement planning, capacity planning, and supply chain planning. I present below two situations I have encountered in my research that require extensions of this model, namely, disassembly planning and planning with component substitutions.

Disassembly planning

The climate crisis is placing pressure on companies to make the most of their end-of-life materials (Özceylan et al., 2019). Many companies have started to recover these materials and disassemble them to salvage components or subassemblies if the item cannot be fixed. However, planning disassembly can be challenging because of uncertainties about the amount of end-of-life items received and the quality of the components. Our two recent studies (Slama et al., 2022, 2024) focus on disassembly planning under lead time uncertainties. In this version of the problem, the disassembly of an item generates components rather than consuming them. Consequently, constraints (2.2) changes to

$$\sum_{\tau=1}^{t} \rho_{i\tau t} r'_{ij} X_{j\tau} + I_{i0} - \sum_{\tau=1}^{t} d_{i\tau} - B_{it} + I_{it} = 0 \qquad i \in \mathcal{I}_e, \quad t \in \mathcal{T}, \tag{2.9}$$

where r'_{ij} represents the number of subassembly i recovered from the disassembly of one unit of j. The constraints (2.3) are modified in a similar manner.

Component substitution

Thevenin et al. (2022a) investigate the performance of the stochastic dual dynamic programming approach for the lot sizing variant with component substitution. In this variant, the manufacturer can switch out a component with another one of better quality to avoid a stockout. The incorporation of component substitution into the BOM provides flexibility, but such substitutions increase costs. Thevenin et al. (2022a) investigate stochastic program-

ming approaches because it would be challenging to manage inventory in this context using standard mathematical analysis methods. To model component substitution, the study introduces variable W_{ijt}, which denotes the amount of item i used to produce j during period t. The consumption is linked to the production with constraints

$$r_{ij}X_{jt} = \sum_{l \in \mathcal{A}_i} W_{ljt}, \qquad j \in \mathcal{I}, \quad t \in \mathcal{T} \tag{2.10}$$

where $\mathcal{A}_i$ denotes the set of items that can substitue i. In addition, constraint (2.3) become

$$\sum_{\tau=1}^{t} \rho_{i\tau t} X_{i\tau} + I_{i0} - \sum_{j \in \mathcal{I}} W_{ijt} - I_{it} = 0 \qquad i \in \mathcal{I}_c, \quad t \in \mathcal{T}. \tag{2.11}$$

Component substitution is an issue commonly encountered during material requirement planning. The lot-sizing model with component substitution applies to supply chain planning, where several sources can supply a node. In particular, models (2.1)-(2.8), (2.10), and (2.11) are generalizations of the model for purchase planning with supplier selection studied in (Thevenin et al., 2022b). The selection of a supplier at a tactical planning level is relevant when the manufacturer has pre-selected a set of suppliers and decides in each period the suppliers to whom he places the order.

2.3 Stochastic programming models in lot sizing

2.3.1 Uncertain parameters in lot sizing

Model (2.1)-(2.8) considers that all parameters of the model are known. However, in practice, several parameters cannot be forecasted accurately, and their variations have a large impact on the decisions and costs. These sensitive parameters vary with the application, and Table 2.2 lists the uncertain parameter in lot sizing, the application context for each parameter, and the reason for the instability of the parameter. Note that the uncertainty in a lot-sizing parameter can be attributed to the variation of a different parameter in practice. For instance, yield uncertainty (the amount of item produced in a lot may differ from what was expected) may be caused by an uncertain production capacity, when manufacturers stop production when they did not have sufficient capacity to complete the lot.

The classical approach to lot sizing assumes that all parameters are deterministic, and safety stock, safety lead times, and safety capacities are calculated separately to account for uncertainty. With advances in computation and optimization, stochastic optimization

approaches can now integrate uncertainty directly into problem formulation.

Although there is a growing amount of research on the LSP under uncertainty (Aloulou et al., 2014; Tavaghof-Gigloo and Minner, 2020), our works (Thevenin et al., 2021, 2022a) are among the first to consider uncertainty in MCLSPs. More precisely, Thevenin et al. (2021) show that stochastic programming leads to significant cost savings for planning under demand uncertainty. Thevenin et al. (2022a) extend this work to account for component substitution, and we propose various optimization approaches. Later, I was involved in the supervision of the PhD thesis of Ilhem Slama, where we studied stochastic programming for disassembly lot sizing under stochastic refurbishing time. More recently, I supervised the PhD thesis of Luo Dan, who studied metaheuristic approaches for large-scale LSPs under uncertainty. The rest of this section presents our contributions to stochastic programming approaches for LSPs.

Our studies focus on the scenario-based approach to stochastic programming. These approaches rely on a set of scenarios, denoted as Ω, where each scenario $\omega \in \Omega$ represents a specific realization of the vector of uncertain parameters. Each scenario has an associated probability p_ω, for which the sum of all probabilities in Ω equals 1. For example, if the demand is uncertain, each scenario ω will provide a value d_{it}^{ω} for the demand of item i in period t. Similarly, if lead times are uncertain, the scenario provides the lead time L_{it}^{ω} for the order of item i passed in period t. It is worth noting that the uncertain yield parameter can model the uncertain lead time is convenient. In such cases, we use parameter $\rho_{i\tau t}^{\omega}$ to indicate the proportion of item i ordered in period τ received in period t. This parameter is nondecreasing with t and allows us to model situations where suppliers make partial deliveries until the full order is fulfilled. In cases where suppliers do not make partial deliveries, $\rho_{i\tau t}^{\omega}$ only takes the values 0 or 1. To model situations where the defects in the production or in the supply, $\rho_{i\tau t}^{\omega}$ may not reach 1. Finally, the available capacity C_{tk}^{ω} and capacity consumption K_{itk}^{ω} can depend on scenario ω.

2.3.2 Different decision framework in lot sizing

A lot-sizing model creates plans for a given horizon $\mathcal{T}$ that commonly corresponds to several years. However, production managers update their plans in every period to account for new information. Therefore, lot sizing applications are often multi-stage decision processes, and stochastic lot sizing formulation can model such a dynamic decision process using a scenario tree. However, these models may be simplified to two-stage processes that ignore the dynamics of the decision process for practical, computational, or methodological reasons.

Table 2.2: Uncertainties in the lot-sizing model: Context or software in which each parameter of the lot-sizing model can be uncertain.

Parameter	**Software/Context**	**Explanations**
Demand	All	Customer demand forecast varies significantly, and some parts of the variations are inexplicable or uncontrollable.
Lead time	Requirement planning	Suppliers are unreliable, and they ship their delivery late.
	Supply chain planning	The transport duration between two nodes of the supply chain varies because traffic conditions are unstable.
	Production planning	The production plan is the input for scheduling. Depending on company policy, the scheduling tool has some flexibility to schedule the job associated with a lot. The actual lead time unfolds after building the schedule.
	Production planning	Poor product quality affects the number of items in a lot. If the company does not ship an order until it is complete, issues with the quality of the product and process may delay the shipment of an order.
	Remanufacturing	The specific operations to refurbish an item depend on the quality of components resulting from disassembly. If the model includes the refurbishing operation in the lead time (rather than as an explicit operation), this lead time is uncertain.
Yield	Production planning	Poor product quality affects the number of items produced in a lot.
	Production planning	Machine breakdown or production delay may result in the impossibility of completing a production lot. In a dynamic decision context, the uncertain capacity translates into an uncertain production yield because the manufacturer observes at the end of the period that they produced a proportion only of the planned order.
Capacity	Production planning	Machine breakdown, employee unavailability.
Capacity consumption	Production planning	Defective item quality may require producing more (or some re-work) items to reach the lot size.
Setup time	Production planning	Technicians often perform setups manually, and their duration may vary significantly.
Setup time/cost	Production planning	Setups require tuning the machines, and technicians often scrap the first produced items. The amount of scrapped items change unpredictability depending on environmental condition (weather, materials, technician, initial conditions of the machine, etc.)

We can distinguish between three decision frameworks: static, static-dynamic, and dynamic. In the static framework, the production planner freezes the plan for the entire decision horizon, and he/she does not update the plan to account for new information. In the dynamic decision framework, the plan is updated in each period to reflect new information. In a static-dynamic framework, only part of the decision is updated in each period. Typically, in the static-dynamic framework, the setups are frozen, and the production quantities are updated in each period. Each of these frameworks is associated with a stochastic program.

The static framework corresponds to a two-stage stochastic model. The model varies with the specific application and types of uncertainty. We provide a generic model based on the formulation of the MCLSP (2.1)-(2.8) given in section 2.2. We provide this formulation for the case where all parameters are uncertain, and the resulting model generalizes several problems I studied recently.

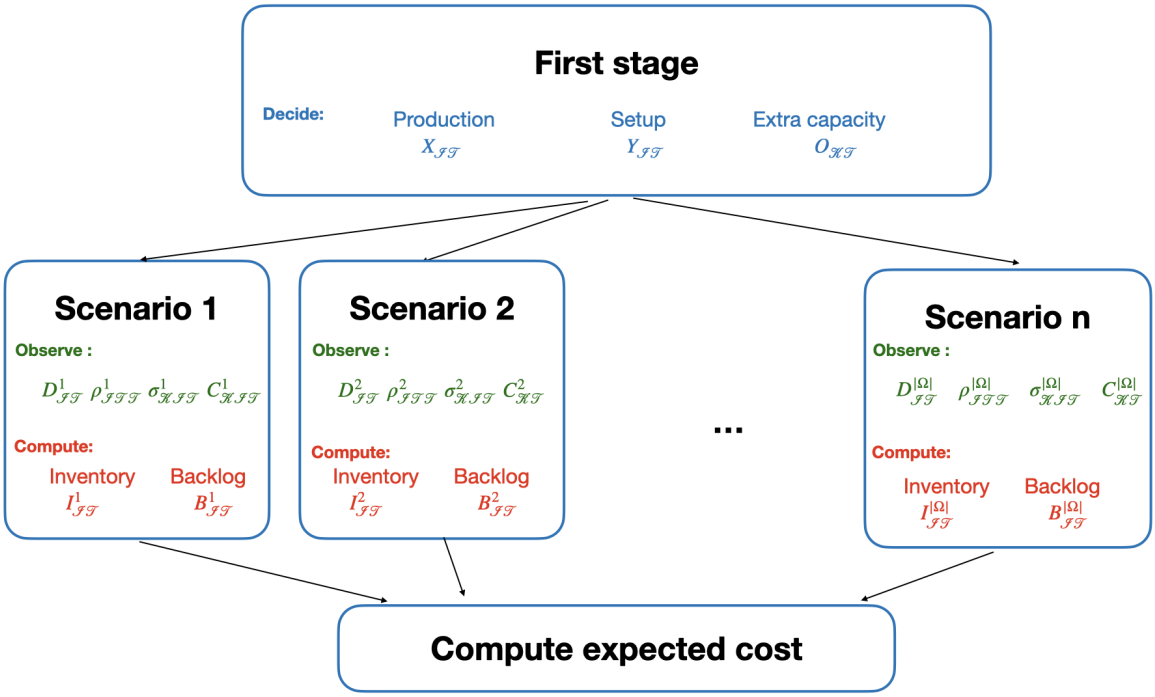


Figure 2.2: Decision process in the two-stage model

$$\min \quad \sum_{\omega\in\Omega}\sum_{t\in\mathcal{T}}\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}(h_i I_{it}^{\omega} + E_i Y_{it} + v_i X_{it}) + \sum_{i\in\mathcal{I}_e}\sum_{t=1}^{t=T-1} b_i B_{it}^{\omega} + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} o_{kt} O_{kt} \tag{2.12}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \rho_{i\tau t}^{\omega} X_{i\tau} + I_{i0} - \sum_{\tau=1}^{t} d_{i\tau}^{\omega} - I_{it}^{\omega} + B_{it}^{\omega} = 0 \qquad i\in\mathcal{I}_e,\quad t\in\mathcal{T},\quad \omega\in\Omega \tag{2.13}$$

$$\sum_{\tau=1}^{t} \rho_{i\tau t}^{\omega} X_{i\tau} + I_{i0} - \sum_{\tau=1}^{t}\left(\sum_{j\in\mathcal{I}} r_{ij}\cdot X_{j\tau}\right) - I_{it}^{\omega} = 0 \qquad i\in\mathcal{I}_c,\quad t\in\mathcal{T},\quad \omega\in\Omega \tag{2.14}$$

$$X_{it} \leq M_i Y_{it} \qquad i\in\mathcal{I},\quad t\in\mathcal{T} \tag{2.15}$$

$$\sum_{i\in\mathcal{I}} \sigma_{ik}^{\omega} X_{it} \leq C_{kt}^{\omega} \qquad t\in\mathcal{T},\quad k\in\mathcal{K},\quad \omega\in\Omega \tag{2.16}$$

$$B_{it}^{\omega} \geq 0 \qquad i\in\mathcal{I}_e,\quad t\in\mathcal{T},\quad \omega\in\Omega \tag{2.17}$$

$$I_{it}^{\omega} \geq 0 \qquad i\in\mathcal{I},\quad t\in\mathcal{T},\quad \omega\in\Omega \tag{2.18}$$

$$X_{it} \geq 0 \quad \text{and} \quad Y_{it} \in \{0,1\} \qquad i\in\mathcal{I},\quad t\in\mathcal{T} \tag{2.19}$$

$$O_{kt} \geq 0 \qquad k\in\mathcal{K},\quad t\in\mathcal{T}. \tag{2.20}$$

In model (2.12)-(2.20), only the variables I_{it}^{ω} and B_{it}^{ω} are scenario-dependent. This means that the decisions on production quantity, setup, and extra capacity are the same for all scenarios. However, for each scenario, we compute the level of backlog and inventory based on the realization of uncertain parameters. The objective is to minimize the expected cost across all scenarios. This decision process is illustrated in Figure 2.2, where the setup, production, and capacity decisions are considered as first-stage variables, and the inventory and backlog levels are computed after observing the demand, yield time, process duration, and capacity in each scenario. The objective is to minimize the expected cost over all scenarios.

The static-dynamic framework corresponds to a multi-stage model. We can represent the decision framework with a decision tree, as shown in Figure 2.3. Each level of the tree corresponds to a decision stage. In the first stage, no information is available, and we decide the setups for the entire horizon, the quantity to produce/order/transport in period 1, as well as the consumption and extra capacity for period 1. At the beginning of period 2, the value of the unknown parameter associated with period 1 unfolds. More precisely, we observe the demand, capacity, capacity consumption, and proportion of each order that arrived. Based on this observation, we compute the inventory and backlog level at the end of period 1 and decide the amount to produce/order/distribute, and the extra capacity in period 2. Several nodes are present in the second stage, and each node represents a possible realization of an uncertain parameter observed at the beginning of stage 2. Decisions are associated with each

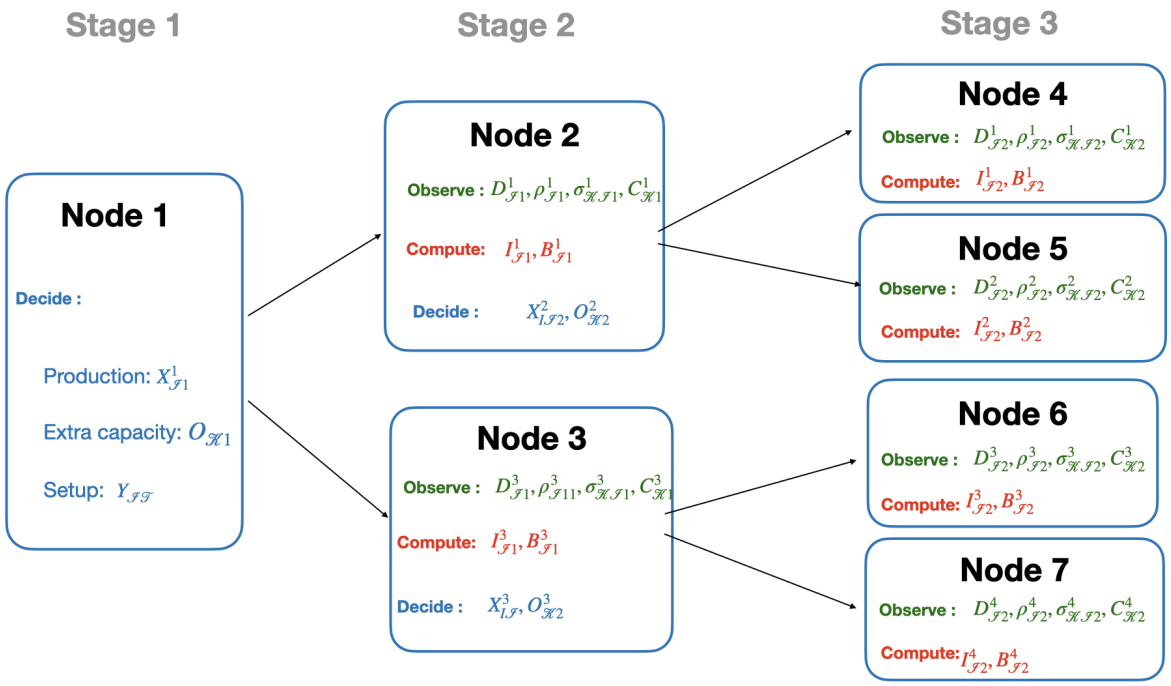


Figure 2.3: Decision Process in the two-stage model

node, and they correspond to the decisions made to react to the observed realization of the parameter in the node. Each path on the tree represents a scenario.

We can translate this scenario tree into a mathematical model. For clarity, we provide the formulation with explicit non-anticipativity constraints. However, a more efficient formulation exists. In the explicit non-anticipativity formulation, the decision made to react to new information that unfolds in each stage depends on scenario (X^ω_{it}, O^ω_{ik}). The non-anticipativity constraints are enforced to make identical decisions at a node of the tree. All scenarios identical up to a node of the tree must lead to the same decisions. If the model does not account for this non-anticipativity constraints, the solver will adjust the solution to the information of the scenario that is not yet revealed in stage t, and thus the decisions are "anticipative". Note that an anticipative solution is not implementable, because it gives several decisions to react to the same information, with no information on which one to implement. In non-anticipativity constraints (2.26) and (2.27), the function $N(\omega, t)$ returns the node of the decision tree associated with scenario ω in stage t.

$$\min \quad \sum_{\omega\in\Omega}\sum_{t\in\mathcal{T}}\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}(h_i I_{it}^{\omega} + e_i Y_{it} + v_i X_{it}^{\omega}) + \sum_{i\in\mathcal{I}_e}\sum_{t=1}^{t=T-1} b_i B_{it}^{\omega} + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} o_{kt} O_{kt}^{\omega} \tag{2.21}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \rho_{i\tau t}^{\omega} X_{i\tau}^{\omega} + I_{i0} - \sum_{\tau=1}^{t} d_{i\tau}^{\omega} - I_{it}^{\omega} + B_{it}^{\omega} = 0 \qquad i\in\mathcal{I}_e, \quad t\in\mathcal{T}, \quad \omega\in\Omega \tag{2.22}$$

$$\sum_{\tau=1}^{t} \rho_{i\tau t}^{\omega} X_{i\tau}^{\omega} + I_{i0} - \sum_{\tau=1}^{t}\left(\sum_{j\in\mathcal{I}} r_{ij}\cdot X_{j\tau}^{\omega}\right) - I_{it}^{\omega} = 0 \qquad i\in\mathcal{I}_c, \quad t\in\mathcal{T}, \quad \omega\in\Omega \tag{2.23}$$

$$X_{it}^{\omega} \leq M_i Y_{it} \qquad i\in\mathcal{I}, \quad t\in\mathcal{T} \tag{2.24}$$

$$\sum_{i\in\mathcal{I}} \sigma_{ik}^{\omega} X_{it}^{\omega} \leq C_{kt}^{\omega} + O_{kt}^{\omega} \qquad t\in\mathcal{T}, \quad k\in\mathcal{K}, \quad \omega\in\Omega \tag{2.25}$$

$$X_{it}^{\omega} = X_{it}^{\omega'} \qquad i\in\mathcal{I}, \quad t\in\mathcal{T}$$
$$\omega,\omega'\in\Omega | N(\omega,t) = N(\omega',t) \tag{2.26}$$

$$O_{kt}^{\omega} = O_{kt}^{\omega'} \qquad k\in\mathcal{K}, \quad t\in\mathcal{T}$$
$$\omega,\omega'\in\Omega | N(\omega,t) = N(\omega',t) \tag{2.27}$$

$$B_{it}^{\omega} \geq 0 \qquad i\in\mathcal{I}_e, \quad t\in\mathcal{T}, \quad \omega\in\Omega \tag{2.28}$$

$$I_{it}^{\omega} \geq 0 \qquad i\in\mathcal{I}, \quad t\in\mathcal{T}, \quad \omega\in\Omega \tag{2.29}$$

$$Y_{it} \in \{0,1\} \qquad i\in\mathcal{I}, \quad t\in\mathcal{T} \tag{2.30}$$

$$X_{it}^{\omega} \geq 0 \qquad i\in\mathcal{I}, \quad t\in\mathcal{T}, \quad \omega\in\Omega \tag{2.31}$$

$$O_{kt}^{\omega} \geq 0 \qquad k\in\mathcal{K}, \quad t\in\mathcal{T}, \quad \omega\in\Omega. \tag{2.32}$$

Two-stage models do not account for the dynamic nature of the decision process. However, they can still provide satisfactory solutions in a dynamic decision context when used in a rolling horizon framework (also known as a receding horizon). The rolling horizon framework optimizes the model in each period, where it considers the new state of the system (inventory levels, on-order quantities, etc.). Since a two-stage stochastic program overlooks the dynamic nature of the decision process, solutions obtained by applying it in a rolling horizon framework can be viewed as a heuristic. However, when the decision in period 1 does not affect the ability of the company to respond to unfolding information in the future, the application of a two-stage model in a rolling horizon yields an optimal solution. For instance, in a single-echelon LSP with zero setup cost, using the two-stage model in a rolling horizon framework is optimal. In this special case, Thevenin et al. (2021) prove that the two-stage and multi-stage models yield the same decisions for period 1. The intuition is that the decision made in period 2 or later can change the state of the system to any value,

and the decisions made in period 1 do not impact the reactivity in period 2. In contrast, in a multi-echelon BOM, the decision in period 1 places the inventory of components strategically in the noes of the supply chain to react efficiently to uncertain demand in future periods.

Rolling horizon frameworks can also embed multi-stage models when these models are approximations that involve sampling. Often, the scenarios considered in stochastic programs are samples from distributions (see section 2.3.3). The value of unknown parameters observed in reality may not be included as part of the sample, which means the actual system state may not be included in the tree. Consequently, the multi-stage model may not provide a decision for the current state. Some approaches suggest using the decision associated with a similar state instead. However, our results suggest that, for lot sizing applications, it is preferable to re-run the multi-stage model to account for the observed state. As information is updated weekly or monthly in lot sizing applications, we can devote a few hours of computation to update the decisions.

We denote by dynamic-dynamic/static-dynamic/static *decision framework* the practical environment that describes how the company operates. We denote by dynamic-dynamic/static-dynamic/static *model* the stochastic program that may not represent the decision process precisely because it can be used in a rolling horizon approach.

Thevenin et al. (2021) show that the use of stochastic programming in material requirement planning reduces the cost significantly (by approximately 20% in our experiments) when compared to the standard approach using a deterministic lot-sizing model with safety stocks. Furthermore, simulations conducted using a rolling horizon framework showed that the multi-stage model resulted in slightly lower costs (around 2% in our experiments) when compared to a two-stage model because of its better placement of safety stock among the components. Thevenin et al. (2022a) extend this work by considering the case where components are substitutable. In this case, manufacturers can replace the required component with a higher-quality one, if the required component is not available for production. We have proven that stochastic programming is valuable in this context since multi-stage stochastic models allow manufacturers to rely on component substitution to reduce risk. The multi-stage program creates a joint safety stock at the component level for all products that use a particular component owing to the dynamic decision-making process.

In our recent work (Luo et al., 2024), we consider the dynamic-dynamic stochastic model for multi-echelon lot sizing. We rely on fix-and-optimize heuristics to solve this challenging problem. In addition, we evaluate the performance of dynamic, static-dynamic, and static models applied in a dynamic-dynamic decision framework. The results indicated that con-

sidering dynamic-dynamic models is crucial when there may be no demand during some periods. For example, our experiments with zero-inflated Poisson distribution showed that dynamic-dynamic models reduced costs by roughly 5% compared to static-dynamic models.

We also looked into the use of stochastic programming to address uncertain lead times in disassembly processes. In (Slama et al., 2022), we consider a case where refurbishing time is stochastic in a remanufacturing facility. The challenge is in planning disassembly processes when the disassembly line is the bottleneck. After the disassembly, components are sent for refurbishing, and they are only available for shipment after a refurbishing time. The approach based on stochastic programming is more generic than classical planned lead time computation such as (Slama et al., 2020) because it can account for the case with several components, capacity constraints, etc. Moreover, in (Slama et al., 2024), we extend the study (Slama et al., 2022) to compute disassembly lot sizes in a multi-echelon bill of materials. We show that embedding static-dynamic models within a rolling horizon frame yields a good heuristic for a dynamic decision-making framework.

2.3.3 Solution approaches

Solving the dynamic or static-dynamic stochastic lot-sizing model is challenging because the number of variables grows exponentially with the number of periods. We investigated several approaches to alleviate this issue. This section successively presents scenario sampling approaches, reformulation of constraints to improve the model, stochastic dual dynamic programming, and fix-and-optimize heuristics.

Scenario sampling

When the set Ω includes all possible realizations of uncertainty, solving program (2.21)-(2.31) yields optimal lot sizes. However, the number of scenarios is often too large (sometimes infinite) to generate or solve the model. To address this issue, a common approach uses an approximation of the model, where the set Ω of scenarios is a sample of the entire set of scenarios. Scenario sampling is necessary when the distribution of the stochastic parameter has continuous support.

Monte Carlo sampling is a simple and efficient scenario sampling approach. To generate a scenario, the Monte Carlo approach draws random values for each uncertain parameter by following the distribution, and all scenarios have the same probability. The basic Monte Carlo sampling is efficient, but advanced scenario sampling techniques (e.g., Quasi-Monte Carlo) can lead to good approximation with fewer scenarios. For instance, the experimental result

in (Thevenin et al., 2021) shows that the QMC method obtains a similar approximation with around four times fewer scenarios. For more information on advanced scenario sampling, we refer the reader to (L'Ecuyer and Lemieux, 2000; Homem-de Mello and Bayraksan, 2015). For multi-stage programs, the scenario tree can be sampled. Our works (Thevenin et al., 2021; Slama et al., 2024; Luo et al., 2022) show that sampling scenarios in a multi-stage context requires a considerably larger set of scenarios. Our experiments suggest that several thousand scenarios are necessary, although computational limitations restrict such experiments to relatively small sample sizes. When sampling a scenario tree, the shape of the tree has a major effect on the performance of the approach (Thevenin et al., 2021; Slama et al., 2024). The number of branches in the tree should be large for the early stage and considerably smaller for the later ones.

One must be cautious with the solution of the approximated model because the costs are biased and represent a statistical lower bound on the true cost. As the solver optimizes over a sample of scenarios, the solution is adapted to the given set of scenarios. Therefore, the resulting costs tend to increase with the number of scenarios because it is easier to adapt to a small number of scenarios. Conversely, true costs tend to decrease with the number of scenarios because a larger number provides a better approximation of the stochastic process. Figure 2.4 illustrates the typical shape of the cost provided by the optimizer and the actual true cost of the solution.

We provide experimental comparisons of the approximated versus true cost in the context of material requirement planning and disassembly planning in (Thevenin et al., 2021) and (Slama et al., 2022), respectively. These results suggest that, for a two-stage stochastic program, 200 scenarios provide a precise approximation of the costs. The true cost in these experiments is obtained by simulation. The evaluation of the cost on the entire set of scenarios is impossible, the true cost is approximated with sampling. The simulation does not optimize decisions, and therefore, the resulting cost is an unbiased estimate of the true cost.

Reformulation of stochastic constraints

Reformulation of the model may considerably reduce the computation time required to solve stochastic programs. For instance, in (Slama et al., 2022), we provide an efficient reformulation of the stochastic lot-sizing model for disassembly planning. We modify the constraints that compute the inventory and the backlog level such that these computations involve fewer unknown parameters.

In this study, we assume that the refurbishing lead time has a finite support of $\left\{\check{L}, \ldots, \hat{L}\right\}$,

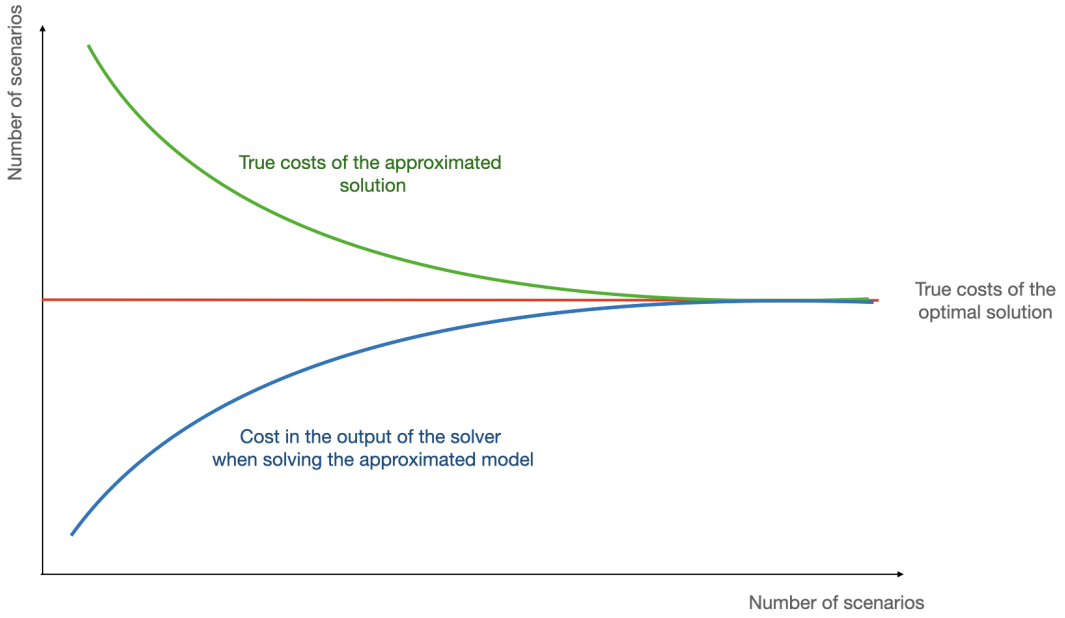


Figure 2.4: Typical shape of the in and out of sample costs depending on the number of scenarios

which is a common assumption from the literature on lead time uncertainties (e.g., Louly et al., 2008). We reformulate the inventory and backlog level constraints (2.9) to limit interactions between different stochastic parameters. Assuming finite support for the lead time implies that $\rho^{\omega}_{j\tau t} = 1$ for all $\tau < t - \check{L}$, and $\rho^{\omega}_{j\tau t} = 1$ for all $\tau > t + \hat{L}$. We observe that the computation of the inventory level in period t requires only considering orders placed in periods in the interval $\left\{t - \hat{L}, \ldots, t - \check{L}\right\}$. We use this property to provide an efficient reformulation of the model. Furthermore, our experiments show that this reformulation has better behavior when the expected costs are approximated with scenario-sampling techniques.

In the two-stage context, the expected inventory level in period t can be computed independently of the inventory level in period $t-1$. As a result, the expected inventory and backlog level for each item and period can be computed based on a different set of scenarios Φ_{it}, as shown in equations (2.33). The objective function can include the term (2.34) that compute the expected inventory and backlog cost independently for each scenario. Decomposing the scenario set per item and period avoids the combinatorial explosion of the number of scenarios with the number of periods. In addition, considering a distribution with finite support significantly limits the number of realizations of the lead time for each item and

period.

$$\sum_{\tau=1}^{t} \rho^{\phi}_{j\tau t} R_{ij} X_{j\tau} + I_{i0} - B^{\phi}_{it} + I^{\phi}_{it} = 0 \qquad i \in \mathcal{I}_e, \quad t \in \mathcal{T}, \quad \phi \in \Phi_{it} \tag{2.33}$$

$$\sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \sum_{\phi \in \Phi_{it}} p_{\phi}(b_i B^{\phi}_{it} + h_i I^{\phi}_{it}) \tag{2.34}$$

Our results (Slama et al., 2022) show that this method can be combined with scenario sampling. The sampling approximation is better in the reformulation because we sample over a smaller number of scenarios. Furthermore, in (Slama et al., 2024), we show that this method can be applied to the static model with a multi-echelon bill of material. However, the approach cannot be applied to the dynamic or static dynamic model with a multi-echelon BOM because the computation of the inventory in a period depends on the inventory values in the previous period.

Stochastic Dual Dynamic Programming

Stochastic dual dynamic programming (SDDP) is a solution approach for stochastic programs that relies on the decomposition of the problem per decision stages. Thevenin et al. (2022a) investigate the performance of SDDP for the lot sizing variant with component substitution.

When dealing with a two-stage problem, SDDP simplifies to the L-shape method. The decomposition produces a first-stage problem alongside a set of subproblems that correspond to the second-stage recourse decision. In the static-dynamic model, the first stage subproblem decides the setup for the entire horizon and quantity to order in the next period. The second stage subproblem computes the level of inventory and backlog depending on the observed demand. There can be one subproblem for each potential scenario in the second stage. Figure 2.5 illustrate the L-shape method for the single period LSP. The first-stage subproblem determines the optimal setup value and production quantity in period 1. The second-stage subproblem calculates the corresponding inventory and backlog levels for each scenario, and each scenario is a realisation of the demand in period 1. The first-stage problem contains an approximation of the expected inventory and backlog costs associated with the first-stage decision (so called "cost to go" function). This approximation is initially the 0 function, and it is built iteratively. More precisely, the approximation of the future cost is a collection of hyperplanes, and SDDP generates these hyperplanes (also known as cuts) iteratively. In each iteration, the first-stage model communicates production quantities and setup values

(e.g., $Y_1 = 1$ and $Q_1 = 5$ in Figure 2.5) to the second-stage model, and the second-stage problem computes the inventory and backlog costs associated with the solution from the first stage, while the second-stage subproblem computes the inventory and backlog costs for each demand scenario. In addition, based on the geometric properties of the dual solution of the subproblem, the L-shape method generates a cut that improves the approximation of future costs. This cut provides the actual costs at the point of the first stage solution, and otherwise, it is an outer approximation.

SDDP extends the L-shape method to multi-stages decision processes. Figure 2.6 shows the decomposition of the static-dynamic model into $T + 1$ subproblems corresponding to the different decision stages. In each stage t, we observe the demand d^{ω}_{it-1} and reception $\rho^{\omega}_{i\tau t-1}$, we compute the level of inventory I^{ω}_{it-1} and backlog B^{ω}_{it-1}. Figure 2.6 assumes the planner decides the production quantity X^{ω}_{it} and extra capacity O^{ω}_{it} at the beginning of period t whereas he/she only observes the realization of the uncertain parameter at the end of a period (thus, they impact decisions made in stage $t + 1$). Finally, in the static-dynamic framework, the setup decision for the entire horizon is made in stage 0 when all uncertain parameters are unknown.

Each subproblem includes an approximation of the future costs that represent the optimal expected costs in period $t + 1$ to T as a function of the decision made in stage t. SDDP builds these outer approximations iteratively. In each iteration, a cut similar to the Benders optimality cut is added to the problem of stage t using dual information on the solution of the problem of stage $t + 1$. The algorithm successively performs forward and backward passes. The forward pass solves the subproblem of stage 1 up to T to generate a feasible solution for the problem. The resulting solution is not optimal because the approximation of future costs may not be of good quality. Therefore, the solution of the forward pass gives an upper bound of the cost of the problem. The backward pass starts in stage $T - 1$, and it generates a cut to the problem with dual information from the solution of the problem of stage T. After generating the cut, the backward pass solves the problem of stage $T - 1$ to create a solution that accounts for the newly generated cut, and it moves to stage $T - 2$. This process continues until stage 1.

For conciseness, this document does not provide detailed models of the entire decomposition. Interested readers are referred to (Thevenin et al., 2022c). As an example of a subproblem in SDDP, I provide the model of the subproblem in stage $t + 1 \in 1, \ldots, T$. In stage t, the system is in a state that depends on the decision made in stage 1 to t, and on the realization of the uncertain parameter in period 1 to $t + 1$. The decisions of the previous

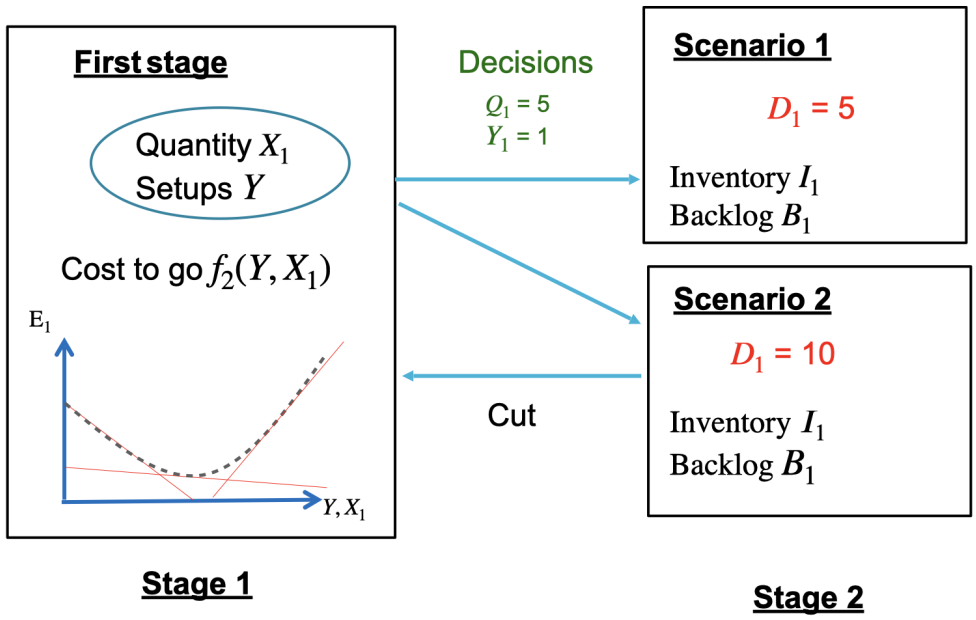


Figure 2.5: Decomposition and steps of the L-shape approach.

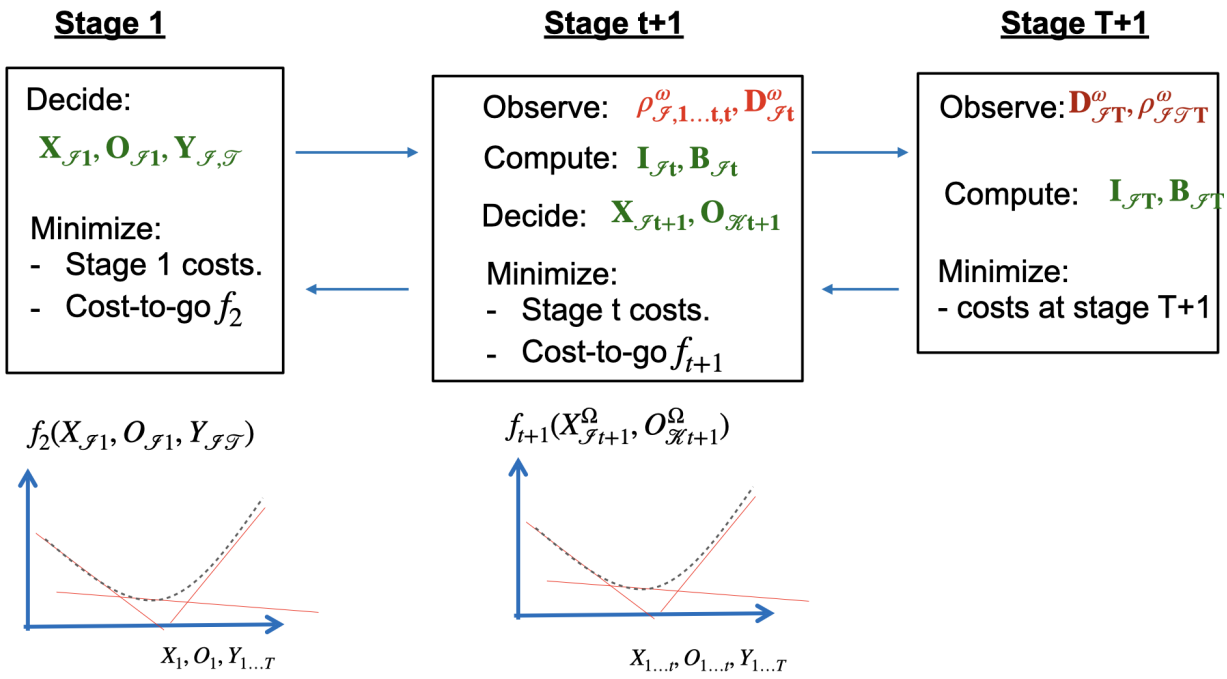


Figure 2.6: Decomposition of the static-dynamic model in the SDDP approach

stages include the setups $\breve{\mathbf{Y}}_{\mathcal{I}\mathcal{T}}$, the order quantity $\breve{X}_{i\tau}$, and the extra capacity $\breve{O}_{i\tau}$ in periods τ equals 1 to t. The realization of the uncertain parameter is a sample path $\xi \in \Xi$, and it gives the demand $d^{\xi}_{i\tau}$ and the proportion of item received for each order $\rho^{\xi}_{i\tau}$. The mathematical model of the subproblem of stage t is as follows:

$$\min \quad \sum_{\omega\in\Omega_t}\sum_{i\in\mathcal{I}}\left(h_i I^{\omega}_{it} + v_i X^{\omega}_{it+1}\right) + \sum_{i\in\mathcal{I}_e} b_i B^{\omega}_{it} + \sum_{k\in\mathcal{K}} o_{kt+1} O^{\omega}_{kt+1} + f_{t+1} \tag{2.35}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \rho^{\omega}_{i\tau t}\breve{X}^{\xi}_{i\tau} + I_{i0} - \sum_{\tau=1}^{t-1} d^{\xi}_{i\tau} - d^{\omega}_{it} - I^{\omega}_{it} + B^{\omega}_{it} = 0 \qquad i\in\mathcal{I}_e, \quad \omega\in\Omega_t \tag{2.36}$$

$$\sum_{\tau=1}^{t} \rho^{\xi}_{i\tau t}\breve{X}^{\xi}_{i\tau} + I_{i0} - \sum_{\tau=1}^{t}\left(\sum_{j\in\mathcal{I}} r_{ij}\breve{X}^{\xi}_{j\tau}\right) - \sum_{j\in\mathcal{I}} r_{ij} X^{\omega}_{jt+1} - I^{\omega}_{it+1} = 0 \quad i\in\mathcal{I}_c, \omega\in\Omega_t \tag{2.37}$$

$$X^{\omega}_{it+1} \leq M_i \breve{Y}_{it+1} \qquad i\in\mathcal{I}, \quad \omega\in\Omega_t \tag{2.38}$$

$$\sum_{i\in\mathcal{I}} \sigma^{\omega}_{ik} X^{\omega}_{it+1} \leq C^{\omega}_{kt+1} + O^{\omega}_{kt+1} \qquad t\in\mathcal{T}, \quad k\in\mathcal{K}, \quad \omega\in\Omega \tag{2.39}$$

$$f_{t+1} \geq \mathcal{F}^{l}_{t+1}(\mathbf{X}^{\mathbf{\Omega}}_{\mathcal{I}\mathbf{t+1}}, \mathbf{O}^{\mathbf{\Omega}}_{\mathcal{K}\mathbf{t+1}}) \qquad l\in\mathcal{L}_{t+1} \tag{2.40}$$

$$I^{\omega}_{it} \geq 0 \qquad i\in\mathcal{I}_c, \quad \omega\in\Omega \tag{2.41}$$

$$I^{\omega}_{it+1}, B^{\omega}_{it+1} \geq 0 \qquad i\in\mathcal{I}_e, \quad \omega\in\Omega \tag{2.42}$$

$$X^{\omega}_{it+1} \geq 0 \qquad i\in\mathcal{I}, \quad \omega\in\Omega \tag{2.43}$$

$$O^{\omega}_{kt+1} \geq 0 \qquad k\in\mathcal{K}, \quad \omega\in\Omega. \tag{2.44}$$

The objective function includes the costs of stage $t+1$ and variable f_{t+1} represents the future costs. Constraints (2.36) and (2.37) compute the inventory level depending on the amount ordered and depending on the realization of the uncertain demand and yield. Finally, constraints (2.38) set the production quantity to 0 if there is no setup, and the capacity constraints (2.39) ensure there are a sufficient number of available resources for production. The future costs are calculated using a set of cuts in (2.40), where $\mathcal{L}_{t+1}$ refers to the current set of cuts in stage $t+1$. The variable f_{t+1} is an outer approximation of the optimal expected costs from period $t+1$ to T based on the decision made in stage t. SDDP builds this outer approximation iteratively, and therefore, the set $\mathcal{L}_{t+1}$ is initially empty, and a cut is added in each backward pass.

Our study (Thevenin et al., 2022c) demonstrates that implementing classical improvements of Benders decomposition results in improved performance for SDDP. These improvements include the use of strong cuts, fast generation of cuts by solving the linear relaxation of the problem in the first phase, and enrichment of the subproblems by retaining the av-

erage demand scenarios. The resulting method can solve problems corresponding to a large scenario tree. For example, a decomposition with 10 scenarios per stage corresponds to a tree with 10^T scenarios. We also propose a heuristic version of SDPP, where the problem is iteratively solved with a fixed setup value, and we re-solve the first stage after convergence to improve the setup.

SDDP cannot be directly applied to models that are not convex because it requires convex subproblems to generate cuts. Therefore, the classic SDDP cannot solve the dynamic-dynamic model. Recently, an extension of the method to an integer program was proposed in (Zou et al., 2019). Another shortcoming of SDDP is that it is limited to a symmetric scenario tree. A slight drawback of symmetric scenario trees is that they do not allow modeling the case where the estimated distribution is updated in each period. A major issue with asymmetric scenario trees is that they provide a poor approximation of the stochastic process. A symmetric tree with 10 scenarios per stage and 10 stages leads to 10^{10} scenarios, but the uncertain parameters of each period are represented with 10 scenarios only. In contrast, in the tree with shape $[10, 10, 10, 1, 1, 1, 1, 1, 1, 1]$, the non-symmetric tree provides 1000 values for the stochastic parameters of periods 3 to 10. Such a non-symmetric tree would therefore provide a better approximation of the uncertain parameter in the last period. Quezada et al. (2022) propose decomposing the problem partially, which may alleviate the issue with symmetric trees.

Heuristics and Matheuristics

Mixed integer programming solvers can efficiently solve the LSP, at least for small and medium size instances. Therefore, for large-scale instances of the LSP that can not be directly solved with a MILP solver, an efficient approach is to iteratively solve a part of the full problem with the MILP. Such approaches are commonly used to solve deterministic LSP, and they include relax-and-fix or fix-and-optimize in the literature (Toledo et al., 2015). However, only a few studies consider their application to stochastic problem (Beraldi et al., 2006).

Thevenin et al. (2021) propose solving the LSP under demand uncertainty with a static-dynamic decision framework in two steps. The first step solves the two-stage model where both the setup and quantity are decided in the first stage for the entire horizon and frozen. The values of the setup decision are taken from the solution of the two-stage model, and they are fixed in the multi-stage model. In the second step, we solve the multi-stage model that corresponds to the static-dynamic decision framework. Our results show that this approach

is efficient for solving the static-dynamic decision framework. In addition, this two-step approach can be used in the rolling horizon framework, which leads to close to optimal solutions. In the rolling horizon framework, the problem is resolved in each period to account for the value of the observed demand. The problem is first solved for the horizon 1 to T. The first stage decision (the setup as well as the quantity to produce in period 1) are implemented. Next, the demand for period 1 is observed, and the problem is solved for horizon 2 to T. This latter model must consider the inventory level and on-order quantity.

To solve the challenging dynamic decision framework, we investigate the use of the fix-and-optimizes approach in (Luo et al., 2024). In the resulting model, setup variables are associated with each node of the scenario tree. Therefore, the number of binary variables grows exponentially with the number of periods, and the problem is challenging to solve. The fix-and-optimize approach iteratively solves the problem where most of the setup variables are fixed, and a few are open for optimization. As a result, each iteration solves a manageable problem with only a few binary variables. In (Luo et al., 2024), we show that period-based decomposition, where the setup variables of successive periods are open simultaneously is efficient. In addition, solving the two-stage model provides a good initial solution. Our experiments show that fix-and-optimize outperforms CPLEX for large-scale problems and large scenario trees. However, the method does alleviate memory issues and does not scale up to large supply chains.

2.4 Robust optimization for lot sizing

This section presents my contribution to the robust optimization methods for LSPs. Robust optimization minimizes the cost against the worst-case scenario. Similarly to a stochastic program, a robust LSP can be formulated with scenarios. In LSPs, uncertainty affects the inventory and backlog cost. The worst-case inventory and backlogs cost is represented by variable H as follows:

$$H \geq \sum_{i \in I} \sum_{t \in T} h_i I_{it}^{\omega} + b_i B_{it}^{\omega} \qquad \forall \omega \in \Omega \tag{2.45}$$

The set Ω of all possible realizations of uncertain parameters may be large, which leads to two difficulties. First, the model may contain a large (eventually infinite) number of constraints. Second, optimizing against the worst case can lead to conservative solutions.

The rest of this section presents the approach to circumvent these issues. Section 2.4.1 discusses the definition of uncertainty set Ω, and the application of robust optimization

on the static-dynamic decision framework. This section also presents the distributionally robust concept that integrates robust optimization and stochastic programming. Section 2.4.2 presents methods to solve robust optimization problems, namely, dualization per constraint and adversarial approach.

2.4.1 Robust optimization models

This section presents the robust optimization model I studied. These studies were conducted during the PhD thesis of Paula Metzker and during the research performed with colleagues from the team MODELIS.

Uncertainty set

To avoid over-conservative solutions, the set Ω must be designed to omit very unlikely scenarios. For instance, it is very unlikely that all suppliers deliver all orders extremely late. Similarly, the demand is very unlikely to be very high in each period of the horizon. A common approach to avoid over-conservative solutions in robust optimization is to define a budget of uncertainty Γ. This budget restricts the total amount of deviation of the uncertain parameter from their nominal value. For instance, in (Metzker et al., 2023a), we consider the single item LSP under uncertain yield, where the yield $\tilde{\rho}_t$ takes a value in $[\hat{\rho}_t, \check{\rho}_t]$, and its nominal value is $\bar{\rho}_t$. The deviation Z_t of the yield from its nominal value is constrained by

$$\sum_{t \in T} |Z_t| \leq \Gamma. \tag{2.46}$$

Expert knowledge can be incorporated into problem-specific definitions of uncertainty sets. We provide such a problem-specific definition of the uncertainty set in (Thevenin et al., 2022b) for a purchase planning problem where a manufacturer decides on dates, quantities, and suppliers of orders for a sensitive component while accounting for the suppliers' delays. We consider suppliers with varying delivery dates, reliability (i.e., lead time range), and prices, and we design a robust model to find inventory and backorder costs under the worst-case scenario in a well-defined set. The basic version of this set contained all delivery date scenarios. However, users can reduce conservatism by defining constraints on the uncertainty set. The constraints can be based on common planning metrics such as the total number of late orders over the planning horizon, number of outstanding late orders in a period, and total tardiness over the planning horizon. We propose defining the uncertainty set based on three budgets. The first limits the number of late orders over the planning horizon, the

second limits the number of late orders in each given period, and the third limits the total lateness over the horizon. The last budget accounts for the number of periods an order is late, whereas the first two only consider if the order is late. The value of the budget can be determined through an analysis of historical data or it can represent the expert knowledge and risk aversion of the decision maker.

Static-dynamic decision framework

To deal with a static-dynamic decision context, the robust optimization community often relies on affine decision rules. In the context of the LSP under uncertainty, the production quantity becomes an affine function of the received quantity and demand:

$$X_{it} = \alpha_{it}^{0} + \sum_{j \in \mathcal{I}} \sum_{\tau=1}^{\tau=t} \sum_{\tau'=1}^{\tau'=t} \tilde{\rho}_{j\tau\tau'} \alpha_{it}^{j\tau\tau'} + \beta_{it}^{j\tau} \tilde{d}_{j\tau} \tag{2.47}$$

In (Metzker et al., 2023a), we compare the performance of stochastic programming and robust optimization for lot sizing. The result shows that robust optimization can hedge against yield uncertainty by building stock, and the decision maker can adjust the conservatism of the solution owing to the budget of uncertainty. Further, we extend our results (Metzker et al., 2023a) in (Metzker et al., 2023b) to provide an adjustable robust model that corresponds to a static-dynamic type of uncertainty. The experimental results from extensive computational experiments show that updating the plan after observing the yield value leads to significant cost savings.

Distributionally robust optimization

Distributionally robust optimization minimizes the expected cost for the worst-case distribution among a set of distributions. Typically, distributions are estimated from data, and the precise distribution is never known exactly. Distributionally robust optimization extends the concept of robust optimization to parameters that define probability distribution. For instance, some distributionally robust optimization approaches apply robust optimization concepts to the mean and the standard deviation of the considered distribution.

In (Metzker et al., 2024), we consider the robust perspective on the stochastic multi-item LSP under yield uncertainty. The objective is to minimize the expected cost for the worst-case distribution, where distributions are taken in a well-defined ambiguity set. We consider the Wasserstein and mean absolute ambiguity sets. The Wasserstein ambiguity set contains all distributions within a given distance of a given distribution. In practice, the given distribution

is often the empirical distribution that corresponds to historical data. We consider the 1-norm to compute the distance between two distributions. The mean absolute ambiguity set considers all distributions with a given standard deviation, and its mean is contained in a box $[\bar{p} - m, \bar{p} + m]$. We show that for these two ambiguity sets, the problem may be reformulated as a mixed integer program. We perform experiments with data from a real case study. We split the data set in two, where the first part is considered historical production yield data, and the second set is used to simulate future production yield and evaluate the performance of the approach. The results show that the distributionally robust optimization model outperforms both the stochastic programming approach and robust optimization in terms of expected costs and the 95^{th} percentile of the cost in a simulation.

2.4.2 Solution approaches

This section provides an optimal policy based on the mathematical analysis of robust inventory management problems, before discussing the two most common approaches to solve the robust optimization problem, namely, dualization per constraint and the adversarial approach. Finally, we provide a short note on the complexity of robust LSP.

Optimal policies for robust inventory management problem

The LSP with setup costs and dynamic demand is a special case of inventory management problems. Because of its combinatorial structure, LSPs are commonly solved with mathematical programming approaches. Inventory management refers to a broader range of problems. Often, the inventory management community relies on mathematical analysis to derive inventory management policies. Such rules provide simple formulas to compute the quantity to order. Our research focuses on the LSP with setup costs and dynamic demand. However, we also derive optimal policies in (Metzker et al., 2023a,b) for inventory management problems when the yield is uncertain. Such policies are easy to understand, and they can be computed with a simple Excel sheet. We further leverage such policies to propose a polynomial-time algorithm for robust LSP under yield uncertainty.

Metzker et al. (2023a) provide the optimal policy for the single-item inventory management problem under yield uncertainty with the following condition: (1) $v_t \leq (\bar{\rho} - \hat{\rho}) b_t, \ \forall t \in T$; (2) budget Γ and the capacity are non constraining; (3) the yield is stationary; (4) there is

no setup cost. This optimal ordering policy is:

$$X_t = \frac{\bar{\rho} d_t + \hat{\rho}\left[\left(\frac{h_{t-1}-b_{t-1}}{h_{t-1}+b_{t-1}}\right)\bar{d}_t - \left(\frac{h_t-b_t}{h_t+b_t}\right)\bar{d}_{t-1}\right]}{\left(\bar{\rho}+\hat{\rho}\left(\frac{h_t-b_t}{h_t+b_t}\right)\right)\left(\bar{\rho}+\hat{\rho}\left(\frac{h_{t-1}-b_{t-1}}{h_{t-1}+b_{t-1}}\right)\right)}. \tag{2.48}$$

In the single-period problem that corresponds to a stationary demand, this formula simplifies to :

$$X = \frac{d}{\bar{\rho}+\hat{\rho}\left(\frac{h-b}{h+b}\right)}. \tag{2.49}$$

In (Metzker et al., 2023b), we provide the optimal policy for the adaptive version of the problem under the following conditions: (1) $\frac{(\bar{\rho}-\hat{\rho})(\bar{\rho}+\hat{\rho})}{K} \leq 2\bar{\rho}$, with $K = \bar{\rho}+\hat{\rho}(\frac{h-b}{h+b})$; (2) budget Γ and the capacity are non constraining; (3) the yield is stationary; and (4) there are no setup costs and no unit production costs. This optimal ordering policy is:

$$X_t = \frac{\bar{d}_t - \sum_{\tau=1}^{t-1}\widetilde{\rho}_\tau X_\tau}{\bar{\rho}+\hat{\rho}\left(\frac{h-b}{h+b}\right)}. \tag{2.50}$$

Dualization per constraint

Dualization per constraint is an efficient approach to solve robust optimization problems. However, it often requires approximating the original problem. For LSP, equation (2.45) cannot be dualized directly. As the approach dualizes the constraints individually, it would require computing the inventory and backlog level of a scenario directly in the same constraint. However, such a constraint would not be linear. Therefore, the worst-case inventory and worst-case backlog are computed separately as follows:

$$H_{it} \geq h_i \sum_{\tau=1}^{\tau=t} \tilde{\rho}_{i\tau t} X_{i\tau} - \tilde{d}_{i\tau} \forall I \in I, t \in T, (\tilde{\mathbf{D}}, \tilde{\rho}) \in \mathcal{U} \tag{2.51}$$

and

$$H_{it} \geq b_i \sum_{\tau=1}^{\tau=t} \tilde{d}_{i\tau} - \tilde{\rho}_{i\tau t} X_{i\tau} \forall I \in I, t \in T, (\tilde{\mathbf{D}}, \tilde{\rho}) \in \mathcal{U} \tag{2.52}$$

where $(\tilde{\mathbf{D}}, \tilde{\rho})$ denotes the vector of uncertain parameters that takes value in the ambiguity set $\mathcal{U}$. The component $\tilde{d}_{it}$ of the vector denotes the uncertain demand for item i in period t, and component $\tilde{\rho}_{i\tau t}$ denotes the random arrival of the produced or ordered items.

In the rest of this section, we present the dualization of constraints (2.51), and the reformulation of (2.52) is achieved in a similar manner. The reformulation with the dualization by

constraint approach requires three steps. First, to avoid the infinite number of constraints, we introduce a maximization subproblem on the right-hand side of the constraint.

$$H_{it} \geq \max_{(\tilde{\mathbf{D}},\tilde{\rho})\in\mathcal{U}} h_i \sum_{\tau=1}^{\tau=t} \tilde{\rho}_{i\tau t} X_{i\tau} - \tilde{d}_{i\tau} \forall I \in I, t \in T \tag{2.53}$$

The second step computes the dual of the right-hand side subproblem. The resulting dual model is a minimization. The third step replaces the maximization problem by its dual. As it is a minimization subproblem, the min sign can be omitted. For more details on the dualization per constraint approach, interested readers are referred to (Metzker et al., 2023a) and (Thevenin et al., 2022b). In (Metzker et al., 2023a), we formulate the robust optimization model for the single item under yield uncertainty. Thevenin et al. (2022b) provide the dualization per constraint for a LSP encountered in the context of purchase planning.

The dualization per constraint approach does not directly apply to most robust optimization problems. In many cases, the approach requires approximating the problem, which results in a model that computes the worst-case cost per constraint rather than globally (Thevenin et al., 2022b). Consequently, the scenario that results in the worst-case cost in period t may differ from that of a different period t'. This approximation yields over-conservative solutions. Another example is the adjustable robust LSP that involves yield uncertainty, which results in a quadratic program that cannot be dualized directly. When the affine representation of X_{it} in equation (2.47) is dropped into Constraints (2.53), uncertain variable $\tilde{\rho}_{i\tau t}$ multiplies other uncertain parameters. Therefore, the subproblem that determines the worst-case value of the uncertain parameter is a quadratic problem that may not be convex and may not have a dual formulation. To circumvent this issue, we suggest approximating the quadratic program by fixing one of the parameters to its minimum or maximum value (Metzker et al., 2023b). Our results show that the approach is highly efficient when compared to an optimal adaptive policy for the robust problem. Additionally, in a simulation environment, this approximation approach leads to less conservative solutions than those of the optimal policy that considers the impact of uncertainty.

Adversarial approaches

The adversarial approach considers two problems, namely, the master problem and the adversarial problem. Figure 2.7 shows the decomposition into two subproblems and the communications between these problems. The master problem corresponds to the original robust

problem, but with a subset only of the scenario. The adversarial problem uses the production quantities given by the master problem as input and returns the worst-case scenario for these decisions. At the start of the algorithm, the master problem contains a single scenario. In each iteration, the master problem generates the current solution $(\mathbf{X}_{\mathcal{IT}}, \mathbf{O}_{\mathcal{KT}})$ that optimizes the cost for the worst-case scenario in the current scenario set Ω. Given this solution, the adversarial problem generates the worst-case scenario ω for the resulting solution. Finally, this scenario ω is added to Ω. The solution of the master problem yields a lower bound of the problem because the worst case is selected in a subset of the scenarios. On the contrary, the adversarial problem gives an upper bound because solution $(\mathbf{X}_{\mathcal{IT}}, \mathbf{O}_{\mathcal{KT}})$ may not be optimal. The algorithm continues until the cost of the worst-case scenario is equivalent to the cost of the master problem, which implies that is upper bound equals the lower bound.

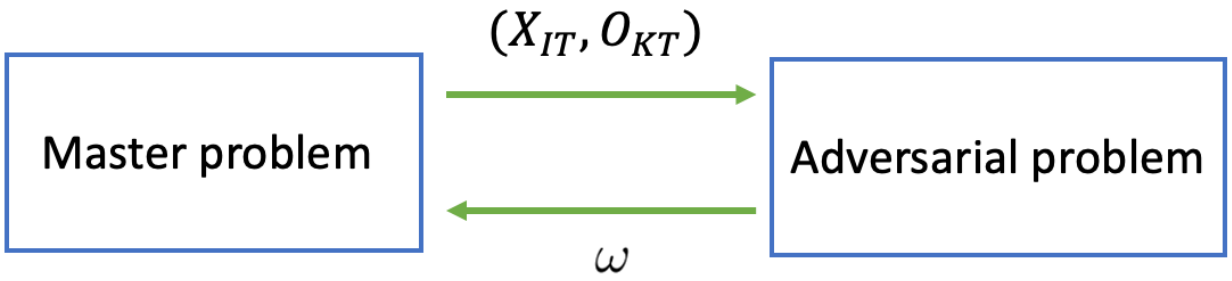


Figure 2.7: Steps of the adversarial approach

In (Metzker et al., 2023b), we consider adjustable robust optimization for LSP under yield uncertainty. The resulting robust model presents a challenge because of its non-fixed recourse, where the uncertain production yield rate is a factor of the decision variable (the production quantity). Such non-fixed recourses prevent the application of the classical reformulation per constraint. To address this issue, we propose an adversarial approach that iteratively generates scenarios to add to the constraint. In this context, the subproblem solves the quadratic subproblem to iteratively generate scenarios to add to the constraint.

Thevenin et al. (2022b) provide an adversarial approach to deal with LSP encountered in the context of supplier selection in a purchase planning application. In this case, the adversarial approach avoids the approximation of the problem, which requires considering the worst case separately for each constraint when applying the dualization per constraint. We observe that the adversarial approach requires a significant amount of CPU time to converge in this application. Therefore, we propose embedding a fix-and-optimize heuristic in the adversarial

approach framework. Algorithm 1 summarizes the basic steps of the approach, where the initial solution is generated from the approximation that applies dualization per constraint. The algorithm iteratively modifies the current solution by fixing most binary variables, and when the algorithm cannot improve the current solution further, the subproblem is solved to identify additional scenarios to add to the problem. Our results indicate that the proposed approach is efficient because it finds close to optimal solutions in a reasonable amount of time.

Algorithm 1: Fix and optimize in the adversarial approach framework

To *generate an initial setup matrix* $\breve{Y}_{\mathcal{TS}}$*:* solve the approximation of the model where dualization per constraint applies.

To *generate an initial set of scenarios:* fix the setup in the robust model to $\breve{Y}_{\mathcal{TS}}$, and solve the model with the adversarial approach. Record the quantities $X^{\star}_{\mathcal{TS}}$ associated with the initial solution.

while *a stopping condition is not met* **do**

  Fix most setup values in model to $\breve{Y}_{\mathcal{TS}}$ and keep a subset open for optimization.

  Solve the resulting model to get the plan $\breve{X}_{\mathcal{TS}}$ with setups $Y'_{\mathcal{TS}}$.

  Compute the cost $f(\breve{X}_{\mathcal{TS}})$ for the worst-case scenario $\breve{\omega}$ with the subproblem of the adversarial approach.

  if $f(\breve{X}_{\mathcal{TS}}) < f(X^{\star}_{\mathcal{TS}})$ **then**

    Set the current solution $\breve{Y}_{\mathcal{TS}} = Y'_{\mathcal{TS}}$.

    Memorize the best known production plan $X^{\star}_{\mathcal{TS}}$.

  else

    Add scenario $\breve{\omega}$ to the set Ω.

  end

end

Complexity results for robust LSP

Robust optimization problems are faster to solve than the corresponding stochastic programming problem. For instance, our results (Metzker et al., 2023a) show that commercial solvers solve the robust single-item LSP under yield uncertainty in less than a second, whereas it takes 15-20 seconds on average to solve the same instances with stochastic programming. In addition, we show that a special case of the robust LSP under yield uncertainty can be solved in polynomial time (Metzker et al., 2023a). In this special case, the production capacity is not constraining, and the maximum and minimum yield values are stationary. However, this result does not generalize to all LSPs. The robust LSPs remain harder to solve than the deterministic problem. For instance, Thevenin et al. (2022b) shows that the robust LSP in

a purchase planning context is NP-Hard, whereas the corresponding deterministic problem can be solved in polynomial time. However, when dualization per constraint applies, the computational time required to solve the LSP model is often close to one of the corresponding deterministic problems. In addition, adversarial approaches tend to converge in a few iterations (Thevenin et al., 2022b).

2.5 Mathematical program for inventory management

The LSP with setup costs and dynamic demand is a type of inventory management problem commonly solved with operations research techniques. In contrast, most inventory management problems are solved using mathematical analysis. One such approach involves expressing the problem as a Markov Decision Process (MDP) and identifying the policy that minimizes the long-run expected costs in the resulting Markov Chain. We employed this approach in (Asgari et al., 2024) to create a policy that jointly optimizes pricing and inventory decisions in a closed-loop assemble-to-order system under lead time uncertainty. Our model assumes that the manufacturer recovers end-of-life items from customers, and the quantity of returns depends on their price. These returned items can provide components for reassembling new items. We assume that the company operates with a base stock policy, where the company orders S_j units of item j to suppliers when the inventory level of the component reaches 0. We derive the optimal order quantity, price of the product, and collection price using mathematical analysis. However, this resulting policy may be limited to specific manufacturing environments and simple policies that enable mathematical analysis. To provide a more generic approach to computing the optimal policy of inventory management problems, our recent work (Thevenin and Adulyasak, 2024) translates the MDP into a linear program that can be solved by commercial solvers, instead of relying on mathematical analysis. This approach allows us to create more scalable models that can be applied to a broader range of manufacturing environments.

Our research work falls within the trend of studies that merge operation research and machine learning or reinforcement learning methods. Most studies in this field seek to boost operation research methods with AI (e.g., Zarpellon et al., 2021; Liu et al., 2022; Bonami et al., 2022; Niroumandrad et al., 2022), or to improve the approach to fit machine learning models (e.g., Blanquero et al., 2023; Carrizosa et al., 2022). There exists a trend in relying on combinatorial optimization approach to solve reinforcement learning problem (e.g., Mazyavkina et al., 2021; Cappart et al., 2021). Our work follows a different direction, where

we rely on a mathematical programming approach to solve coupled constrained MDP (Puterman, 2014). The novelty of our work is the study of these models for application to jointly manage the inventory of several items.

Thevenin and Adulyasak (2024) propose a method for computing inventory management policies for multiple items when joint constraints are imposed on these items by the factory manager. These joint constraints can include a target service level or a maximum inventory constraints. We model this problem with coupled MDPs, where each MDP represents the dynamic inventory management for one item. The state in the MDP corresponds to the inventory level (p), and the action corresponds to the order quantity (q). Therefore, the approach requires discretizing the inventory level into discrete levels in set $\mathcal{P} = \{1, \ldots, |\mathcal{P}|\}$. We also denote by $Q(p)$ the set of quantities than can be ordered when the inventory position is p. These MDPs are linked by joint resource constraints, and linear programming is a common approach for solving such coupled constrained MDPs (Ross and Varadarajan, 1991). To translate the MDP as a linear program, we define a decision variable x_{pq}, which represents the probability of ordering quantity q when the inventory level is p. The cost function for each action state is represented by cost c_{pq}. The transition probability $p(p \mid p', q)$ indicates the probability of moving to state p after being in state p' and ordering q units. The cost and transition matrix can be used to model various situations, such as including costs associated with stockouts, backorders, or other inventory-related costs. The resulting linear translation of the MDP for a given model is as follows:

$$\min \sum_{p \in \mathcal{P}} \sum_{q \in Q(i)} c_{pq} V_{jq} \tag{2.54}$$

$$\sum_{q \in Q(p)} x_{pq} - \sum_{p' \in \mathcal{P}} \sum_{q \in Q(p')} p(j \mid p', q) x_{p'q} = 0 \qquad \forall p \in \mathcal{P} \tag{2.55}$$

$$\sum_{p \in \mathcal{P}} \sum_{q \in Q(p)} x_{pq} = 1 \tag{2.56}$$

$$x_{pq} \geq 0 \quad \forall p \in \mathcal{P}, \forall q \in Q(p). \tag{2.57}$$

The objective function (2.54) minimizes the expected annual total cost associated with inventory decisions. Constraints (2.55) ensure the probability flow conservation between inventory states. These constraints map the state space based on the given transition function derived from demand distribution, inventory state, and replenishment decision. The total probability flow associated with inventory state i is equal to the sum of all probabilities as-

sociated with possible inventory state i and ordering quantity q which can result in inventory level p after the lead time. Constraints (2.56) ensures that the total probability flow sums up to 1. Finally, constraints (2.57) ensure the non negativity of decision variables x_{pq}. Note that the solution from the MDP provides a state-decision mapping function (i.e., it provides quantity q, which corresponds to inventory p).

We extend model (2.54)-(2.57) to the multi-item case under a joint target level. In this context, parameters and variables are indexed with item i, and the model includes a joint resource constraint of the form:

$$\sum_{i \in I} \sum_{p \in \mathcal{P}} \sum_{q \in Q(p)} \phi_{pq}^{li} V_{pq}^{i} \leq R^{l} \quad \forall l \in L. \tag{2.58}$$

Constraints (2.58) can model a wide variety of requirements including common target levels used in practice. For instance, to force the average no stockout probability to be larger than α, $\phi_{pq}^{li} = -P(\tilde{d} \leq p)$ and $R^{l} = -(1-\alpha)$

Although the model is appealing for its generality, it has two main drawbacks. First, the model becomes excessively large when there is a large number of possible values for the inventory level. Second, the resulting policy is a table look-up policy, which may be impractical when compared to inventory management rules such as (s, S) policies. Additionally, the policy may not be deterministic, which means that for a given inventory level, the planners muse select the order quantity by rolling a dice.

To address these issues, we propose several improvements to the model. First, the linear program can be extended with a few constraints to enforce a deterministic policy. Although this extension is straightforward, it results in a mixed-integer linear program that is considerably harder to solve than the original linear program. To avoid scalability issues, Thevenin and Adulyasak (2024) suggest a branch and price algorithm, where the problem is decomposed per item and solved with column generation, where each column represents a possible policy. Second, instead of solving the linear program to generate the policy (column), we propose a polynomial time algorithm to find the optimal (s, S) policy. Finally, we suggest using a fast and efficient algorithm that employs a local search to optimize the (s, S) policy in each iteration of column generation.

One advantage of our approach is that it can handle a wide variety of probability distributions. Our results indicate that the zero-inflated negative binomial distribution outperforms the empirical distribution for slow-moving items, which is consistent with previous literature. In addition, our findings suggest that learning complex policies requires a more precise

estimation of the distribution of the uncertain parameter. For instance, an (s, S) policy is relatively simple because it involves only two parameters to learn, and in this case, the empirical distribution provides good results. In contrast, a table lookup policy is more complex because it requires learning the action associated with each state. Our results indicate that a table lookup policy requires a precise estimate (e.g., with zero-inflated negative binomial distribution) of the probability distribution to perform well.

2.6 Future works in lot sizing under uncertainty

Over the last few decades, a growing amount of researchers have focused on LSPs under uncertainties. However, there remain several research gaps in the literature, and I provide below a few avenues for future research on this topic.

2.6.1 Extension of the mathematical formulation

This section discusses future research directions that require extending the formulation of the LSP to cope with more practical applications. We successively discuss the consideration of other uncertain parameters than demand and lead time, studies where the distribution of the uncertain parameter may be controlled, extensions to account for customer service level, and extensions to operate resilient supply chains.

Modeling different uncertain parameters

Most research on stochastic lot sizing focuses on demand uncertainty, and a limited number of publications consider lead time uncertainties. However, the LSP involves parameters that may vary significantly, such as production yield, production capacity, setup duration, and process duration. Imperfect forecasts of these parameters can affect the production plan performance negatively. In addition, to the best of our knowledge, there exist no studies on the uncertain bill of process. Such studies may be important to manage remanufacturing facilities, where depending on the condition of an item, different operations may need to be implemented. Such work could build upon our results on yield uncertainty and flexible bill of materials.

Studies must investigate if the approach proposed in the previous section directly works for different uncertain parameters. For instance, the two-step fix-and-optimize approach proposed in (Thevenin et al., 2021) solves stochastic models efficiently when setup decisions have

few impacts on the reactivity of the supply chain. There is a need to investigate if the approach remains efficient when parameters besides demand are uncertain.

For robust optimizations, our works investigate yield and lead time uncertainties, whereas most of the literature considers demand uncertainty. Our investigations show that yield and lead time uncertainties lead to challenges that are not encountered when considering only demand uncertainty. For instance, yield uncertainty leads to non-fixed recourse in the adjustable robust optimization framework, and problem-specific methods must be designed to handle this situation. In addition, dualization per constraint does not directly apply to lead time uncertainty because the parameter takes discrete values. Future works must assess the impact of relaxing integrality constraints when computing worst-case lead time. There is also a need to study adjustable and distributionally robust optimization under lead time uncertainty. The discrete supports of lead time distributions in LSP provide a specific structure to the distributionally robust optimization problem, which could lead to interesting discoveries.

Extension to deal with decision-dependent distributions

Most works on stochastic lot sizing assume simple distributions. In practice, the distributions for different parameters are often correlated (e.g., demands for different items are correlated). An avenue for future work is to consider such more complex joint distributions. While the scenario-tree-based methods proposed in section 2.3 should apply directly to the case where distributions are not independent, this situation must be validated based on experimental results. Further, the experiments must show the importance to account for these correlated distributions.

There is also a need to develop methods that can tackle cases where decisions may change the distribution. A few researchers considered the impact of pricing decisions on demand distributions. However, more work is required on this topic. For instance, there is no work on the joint optimization of the decisions of production, dynamic pricing, and placement of orders to suppliers in assemble-to-order systems. This optimization problem is especially relevant for high-tech producers because the components are often substitutable, and a good pricing strategy allows to increase in the demand for products made of components available in large quantities in the stocks. Besides the case where decisions on prices impact the demand, there are other cases where decisions may impact the distribution of the parameters. For instance, our work (Thevenin et al., 2022b) considers supplier selection decisions to change the distribution of the lead time. The lead time distribution may also be influenced

by quantities ordered from the suppliers or the quantity released to the shop floor. Another example is the yield distribution which may depend on the maintenance activities and choice of suppliers of the raw materials.

One approach to address decision-dependent distributions is to compute the value of the uncertain parameter with a linear formula of the decisions. For example, the demand for scenario ω can be calculated as $d_{it}^{\omega} = \beta_{it}^{\omega} - \alpha_{it}^{\omega}\pi_{it}$, where π_{it} is the price of item i in period t, and β_{it}^{ω} and α_{it}^{ω} are parameters of the linear. These parameters may themselves be uncertain and depend on the scenario. Dealing with decision-dependent lead time or yield creates additional difficulties for problem formulation. For instance, the yield multiplies production quantities, and therefore considering the case of controllable yield results in a quadratic model that requires linearization to solve with a MILP solver or an approach to handle quadratic programs.

Chance constraint formulation

Our work focuses on cases where the model seeks to balance inventory and backlog costs under uncertainty. A different paradigm commonly encountered in practice is to minimize inventory costs while respecting the service level. There exist various definitions of the service level, and they lead to different formulations. In such a case, the stochastic program becomes a chance constraint formulation, where the constraint takes the form:

$$\mathcal{P}(\sum_{\tau=1}^{t} X_{it} \geq \sum_{\tau=1}^{t} d_{it}) \geq \alpha \tag{2.59}$$

Such formulations are often nonlinear. Some approaches (e.g., Helber et al., 2013) exist to linearize the constraint when the distribution is smooth and has few parameters. However, linearization remains a challenge when the distribution function is non smooth, or when it involves a large number of parameters. Such situations are for instance encountered when the process duration is uncertain.

Extension of our works to deal with resilient supply chains

Recent events such as the COVID-19 pandemic pushed companies to improve the resilience of their supply chain. Improving the resilience of a manufacturing network requires designing and managing the value chain such that recourse actions are available for most disruptions. The resilience of the supply chain requires carefully selecting suppliers, positioning production sites, positioning the distribution centers, dimension safety stocks, etc. Tackling these

challenges calls for extensions of the models proposed in section 2.3, 2.4, and 2.5.

The model considered in (Thevenin et al., 2022a) is applicable to supply chain planning where a node can receive supplies from different nodes. Such models are relevant in the management of resilient supply chains, where the supply chain must continue operating even when a node is down. Therefore, future work includes extending the model where capacity is uncertain, in a way that the capacity may reach 0 at a node for several successive periods, and this corresponds to a node in disruption. Similarly, suppliers may not be available during several periods. There is also a need to extend the models in (Thevenin et al., 2022a) to provide robust decisions when no data is available (e.g., rare events). To cope with situations where data is missing, robust optimization can provide a theoretical guarantee of the resilience of the supply chain while accounting for the risk aversion of the decision maker. Although being able to handle any disruptive event is appealing, robustness and resilience come with a cost. The right decisions must balance cost and resilience.

In addition, the integration of network design and supply chain planning is a relevant avenue for future research. The resulting problem involves two levels of decisions. The first level designs the supply chain (selection of supplier, placement of the manufacturing facilities, and placement of distribution center), and the second level evaluates the performance of operating with such a supply chain when disruption occurs.

2.6.2 Using historical data to improve planning approaches

Estimation of distribution and validation on simulation

Most work in scenario-based stochastic programming approaches for lot sizing assumes a probability distribution is given. In practice, these scenarios are sampled from probability distributions, and such distributions are estimated only from historical data. I believe that important findings may come from the validation/simulation of the planning approaches proposed in our research with historic data. In such a context, the historical data set is split into past data used to optimize the plan and future data for the simulation.

Stochastic programming approaches often require the use of stochastic forecasting methods to estimate probability distributions from historical data. These distributions are then input into a sampling approach, which generates scenarios or scenario trees. Finally, these scenarios are input into stochastic programming to derive a plan for future decisions. These approaches are used sequentially and come from different communities, such as statisticians, mathematical analysts, and operations researchers. However, only a few works have consid-

ered end-to-end applications. Studying the entire sequence of methods, starting from data, building probability distributions, generating scenarios, and making decisions have led to interesting findings. For instance, we observed that when there is insufficient data to estimate the distribution precisely, inventory management need to rely on a simple policy with few parameters to fit the data. More complex policies are advantageous when more data is available. Our experiment on an inventory control problem showed that an (s, S) inventory management rule outperforms complex non-deterministic table lookup policies when the uncertain distribution is built with only six months of historical data (Thevenin and Adulyasak, 2024). However, when four years of data are available, the table lookup policy outperforms the simple (s, S) policy. These observations call for further study, especially for lot sizing with dynamic demand. The important research question to answer is whether a simple model, based on safety stock computation, would perform better than a complex multi-stage stochastic lot-sizing model when there is insufficient data to estimate the distribution accurately.

In the context of uncertain lead time, an interesting project would be to connect the production planning tool with a discrete event simulator. During production, the production plan serves as input for a production scheduler, which transforms production lots into jobs to perform on machines. Each job in the production scheduler has a release date corresponding to the period in which the lot is planned. Connecting the simulator and production planner would generate data to learn the lead time distribution. The simulator uses the production plan as input and outputs the execution of the plan as a production schedule. We can infer the actual lead times from the simulation, and this data can help learn the distribution of the lead time. The lead-time distribution may be conditional on the production quantity or the execution strategy for the simulator (e.g., earliest due date).

Model acquisition to learn uncertainty sets

While stochastic programming uses a probability distribution for each parameter as the input, robust optimization requires the definition of uncertainty sets. Learning uncertainty sets from data is another avenue for future research. Polyhedral uncertainty sets are interesting because the dualization per constraint approach applies. For instance, Beldiceanu et al. (2020) propose an approach to generate sharp bounds on the values of a time-series parameter. Future works must investigate the relevance of using this approach to generate time-series constraints that define the uncertainty set in robust optimization. The idea is to create a set of constraints such that all (or most) realizations of the uncertain parameters

in the historic data respect these constraints. These constraints must generalize to future realization, and they must be learned with a statistical learning approach to remove outliers. In addition, some work is required to extend the bounds to the case where constraints link multiple time-series. As a result, an analysis of datasets may help in generating complex constraints linking several parameters. Such constraints may uncover relations between the distribution of several parameters (e.g., correlated demands, and demand and lead time).

Distributionally robust optimization

DRO is a young field of research that has been gaining momentum over the past decades. The last thesis chapter of Paula Metzker was the first work on the application of distributionally robust optimization methods to LSPs under yield uncertainty. In this work, we consider the multi-item LSP, where the different items are linked by the ambiguity set. Due to time constraints, we did not investigate the solution approaches. However, we discussed about a decomposition approach for the uncapacitated version of the problem. The idea is to decompose the multi-item lot-sizing model per item, and a mechanism links the different subproblems to ensure the distribution is in the ambiguity set.

Another avenue for future research is the study of distributionally robust optimization for the LSP under lead time uncertainty. The lead time distributions has discrete support. We can investigate approaches similar to the adversarial approach that iteratively generates the worst lead time distribution until convergence. The main benefit of the approach is that it does not require strong assumptions that ensure the tractability of the methods.

2.6.3 Solution approach for large-scale problems

Solving large-scale multi-stage stochastic programs remains a challenge. This section presents several avenues of research on heuristics to deal with large-scale LSP under uncertainty.

Heuristics for scenario tree approaches

Section 2.3.3 presents various approaches to solve scenario tree-based models for the LSP when dealing with large scenario trees. First, we highlight the importance of incorporating an optimization method in a rolling horizon approach. Second, we propose an efficient approach that fixes setups to the solution of the two-stage model approach, before solving the scenario tree model using only continuous variables. Third, we attempt to solve large instances of the stochastic LSP in a dynamic decision framework with a fix-and-optimize approach. These

approaches can be further extended in several ways.

For the multi-stage model that corresponds to a static dynamic decision framework, the two-step approach could be improved by solving the second step with the progressive hedging decomposition approach. Progressive hedging can solve large scenario trees because it decomposes the problem per scenario, and it yields optimal results when the problem is convex. Progressive hedging can also exploit parallel computing. In addition, we could investigate an iterative method where the setup vector is modified in each iteration in a local search framework.

We must continue to explore scenario sampling approximation for multi-stage models. Our results suggest that a tree with more branches in the first stage yields better approximations. Further studies are necessary to evaluate the impact of approaches that approximate the end of the planning horizon with a two-stage model (which would correspond to a truncated scenario tree).

In the context of the thesis of Dan Luo, we are investigating the possibility to integrate the fix-and-optimize method in a variable neighborhood search, to solve the dynamic-dynamic model. The idea is to add random perturbations to the solution to create diversification. We are also investigating if the machine learning method can help select the part of the plan that need to be re-optimized.

Metaheuristics for stochastic dynamic LSPs

To avoid the drawback of the scenario tree size, we explore alternative approaches that do not require such a tree. The use of an affine policy, commonly applied in robust optimization, can transform a multi-stage problem into a two-stage problem. Future work must investigate the design of policies to solve multi-stage stochastic programs. Lot-sizing models can be solved with an (s, S) policy, which provides a rule to update the order quantity based on the current inventory level of each item. The order quantity required for each item in each period can be obtained using the formula:

$$X_{it} = \begin{cases} 0 & \text{if } I_{it} \leq s_{it}, \\ S_{it} - I_{it} & \text{otherwise} \end{cases} \tag{2.60}$$

The values of s_{it} and S_{it} are parameters of the policy, and solving the multi-stage model reduces to finding values for these parameters. However, the (s, S) policy may not be effective when the problem involves additional constraints. For example, when several nodes of a

supply chain require a component with limited availability, the policy must prioritize the consuming nodes. The (s, S) policy can be extended to deal with additional parameters. However, further research is required to design efficient rules. Generally, there is a need to study what type of policies can provide a good approximation of optimal policies, as well as how to learn the necessary parameters.

Approach from the reinforcement learning community

Dynamic stochastic decision problems have been extensively studied in the reinforcement learning community. To tackle the stochastic LSP in a dynamic decision framework, future research needs to explore potential synergies between the AI and operations research communities. Techniques such as neuro-dynamic programming or linear program resolution with machine learning-based cost approximation may be employed to address this situation. Additionally, deep reinforcement learning may be directly applied to solve the stochastic LSP under the dynamic decision framework.

Chapter 3

Assembly line design under uncertainty

This section presents my contribution related to the assembly line design problem, as well as the perspectives and future works on this topic.

3.1 Assembly line design problem

Assembly lines are commonly used at the final step of a mass production process for complex products such as electronic goods, automobile, etc. The assembly steps join several parts produced in different factories. The efficiency of an assembly line relies on breaking down the work into several steps, where each step is carried out at different stations. In the simplest setting, the items flow through the line at regular intervals called takt time. Each item passes through each station, and a finished item exits the line at each takt. The takt time is important because it defines the throughput of the line. Nowadays, with mass customization, each assembly line can assemble an entire family of products. Our works focused on the mixed model assembly line, where the line is designed such that the setup required between two products is negligible.

The assembly line design problem includes decisions such as the number of stations, assignment of tasks to stations, assignment of equipment to stations, and the selection of workers' profiles, among others. The input of the problem is the family $\mathcal{I}$ of products to assemble on the line. For each model variant i, we are given the set of tasks $\mathcal{O}_i$ and $\mathcal{O} = \cup_{i \in \mathcal{I}} \mathcal{O}_i$, durations of the tasks, and required pieces of equipment. The required equipment is defined with parameter R_{oe}, which equals 1 if task o requires equipment e, and 0 otherwise.

The assignment of tasks to stations must respect precedence relations, and we denote by $\mathcal{A}$ the set of precedence relations (o, o'). For more information on the assembly line design problem, we refer interested readers to (Rekiek et al., 2002; Becker and Scholl, 2006; Boysen et al., 2008).

While there exist variants of the assembly line design problem, our works have focused on the case of collaborative assembly. In collaborative assembly, several workers/robots perform tasks at a station. There is growing attention on collaborative assembly owing to the arrival of cobots and mobile cobots in assembly lines (e.g., Sikora and Weckenborg, 2022; Michalos et al., 2018). The research on collaborative assembly builds on the multi-manned assembly line balancing problem (Hashemi-Petroodi et al., 2021, 2020). In particular, our work was inspired by the problem encountered in the automobile industry described in (Battaïa et al., 2015). In this context, there are between 1 and $\hat{l}$ workers in each station, and the duration p^l_{io} of each task o depends on the number of workers l. The objective is to ensure that the line achieves the desired throughput at minimum costs. The cost of the line includes the cost c_e of equipment e and the cost per worker α. The throughput is defined by the takt time C. Our study extends the work (Battaïa et al., 2015; Delorme et al., 2019) to consider that besides workers, the assignment of tasks to stations may change in each takt.

In classical assembly lines, workers are fixed at the station, and they perform the same set of tasks throughout the day. Opposite to such static lines, our recent works (Hashemi-Petroodi et al., 2024) showed that moving workers yield more efficient assembly lines. Worker movements can take different forms, which can help adjust the capacity of the stations to the load (Hashemi-Petroodi et al., 2021). In multi-manned assembly lines, moving workers commonly move from one station to the next. Some lines rely on utility workers reserved to help on stations with a heavy load. In this context, the sequence of items that enter the line impacts the throughput. For instance, if several items with heavy workloads succeed each other in the production sequence, more workers are required to meet the takt time. To avoid extreme cases that might lead to over-conservative line design, we consider a set of possible sequences Ω that contains all production sequences that respect the user-defined constraints (e.g., each sub-sequence of 5 items may not contain variant A more than 3 times).

In the following section, we introduce the assembly line design problem as it is discussed in (Hashemi-Petroodi et al., 2022). This particular variant of the problem assumes that workers are identical and can move freely between stations at the end of each takt. This version serves as an introduction to the problem, and we discuss several other variants in the rest of the chapter. We study robust optimization approaches as well as the use of constrained

MDPs for these problems. A few studies consider robust optimization in assembly line (e.g., Hazır and Dolgui, 2013; Moreira et al., 2015). However, most of these works consider process duration uncertainty, and we are the first to consider robust optimization to design a line that can cope with any incoming sequence of products. While there exists a few works on reinforcement learning for assembly lines, these works focus on specific assembly task (e.g., Li et al., 2022), or they use reinforcement learning to solve the classical assembly line balancing problem (e.g., Tuncel et al., 2014). To the best of my knowledge, we were the first to consider an MDP approach to assign workers and tasks dynamically in line balancing.

The assembly line design problem with moving workers involves making several decisions. These decisions include determining the number of workers to hire (Y), equipment assigned to each station (represented by the binary variable w_{se}, where 1 indicates that equipment e is assigned to station s), assignment of task to stations (represented by the binary variable x_{so}, where 1 indicates that task o is assigned to station s), and assignment of workers to stations (represented by the binary variable $b_{sl}^{\omega t}$, where 1 indicates that there are l workers in station s during period t for sequence ω). In this problem, we use sets $\mathcal{S}$ and $\mathcal{E}$ to represent the set of stations and equipment, respectively. From each sequence ω, we can infer the station s_{it}^{ω} where model i is processed in takt t, as well as the item i_s^t in station i in takt t. The integer linear formulation of the assembly line design problem is as follows:

$$\min \quad \alpha\, Y + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} w_{se} c_e \tag{3.1}$$

s.t.

$$\sum_{s\in\mathcal{S}} \sum_{l=1}^{\hat{l}} l\; b_{sl}^{\omega t} \;\le\; Y \qquad \omega\in\Omega,\; t\in\mathcal{T} \tag{3.2}$$

$$\sum_{l=1}^{l=\hat{l}} b_{sl}^{\omega t} \;=\; 1 \qquad \omega\in\Omega,\; t\in\mathcal{T},\; s\in\mathcal{S} \tag{3.3}$$

$$\sum_{s\in\mathcal{S}} x_{so} \;=\; 1 \qquad o\in\mathcal{O} \tag{3.4}$$

$$b_{oil}^{\omega t} \;\le\; x_{so} \qquad l\in\{1,\dots,\hat{l}\},\; \omega\in\Omega,\; o\in\mathcal{O},\; i\in\mathcal{I},\; t\in\mathcal{T},\; s=s_{it}^{\omega} \tag{3.5}$$

$$b_{oil}^{\omega t} \;\le\; b_{sl}^{\omega t} \qquad l\in\{1,\dots,\hat{l}\},\; \omega\in\Omega,\; o\in\mathcal{O},\; i\in\mathcal{I},\; t\in\mathcal{T},\; s=s_{it}^{\omega} \tag{3.6}$$

$$b_{oil}^{\omega t} \;\ge\; b_{sl}^{\omega t} + x_{so} - 1 \qquad l\in\{1,\dots,\hat{l}\},\; \omega\in\Omega,\; o\in\mathcal{O},\; i\in\mathcal{I},\; t\in\mathcal{T},\; s=s_{it}^{\omega} \tag{3.7}$$

$$\sum_{o\in N_i} \sum_{l=1}^{\hat{l}} p_{io}^{l}\; b_{oil}^{\omega t} \;\le\; C \qquad \omega\in\Omega,\; i=i_s^{t\omega}\; t\in\mathcal{T},\; s\in\mathcal{S} \tag{3.8}$$

$$x_{so} \;\le\; \sum_{e\in E} R_{oe} w_{se} \qquad o\in\mathcal{O},\; i\in\mathcal{I},\; s\in\mathcal{S} \tag{3.9}$$

$$\sum_{s\in S} s\; x_{so} \;\le\; \sum_{s'\in S} s'\; x_{s'o'} \qquad (o,o')\in\mathcal{A} \tag{3.10}$$

$$x_{so},\; b_{sl}^{\omega t},\; w_{se}\in\{0,1\},\; Y\ge 0,\; b_{oil}^{\omega t}\le 1 \tag{3.11}$$

The objective (3.1) is to minimize the costs of the assembly line, which include both worker and equipment costs. Constraints (3.2) ensure that the variable Y is set to the maximum number of workers required across all sequences and takts. In addition, constraints (3.4) guarantee that each task is assigned to exactly one station, while constraints (3.3) determine the number of workers needed at each station during each takt and sequence.

The value of binary variable $b_{oil}^{\omega t}$ is computed from the task assignment x_{soi}^{ω} and the number of worker $b_{sl}^{\omega t}$ in stations with constraints (3.5), (3.6), and (3.7). Equations (3.8), (3.9), and (3.10) define the classical takt time, equipment, and precedence constraints, respectively.

The problem does not model the specific movements of each workers on the line or the specific assignment of tasks to individual workers. The assembly line design problem is a tactical model, and these simplifications make the problem easier to solve. The main objective is to position the equipment on the line and dimension the workforce. The detailed

schedule in each station and the movement of workers are computed at the operational step.

3.2 Worker movement and dynamic task assignment

Flexible task assignment is possible because of new resources such as programmable machines (with computer numerical control) and programmable robots/cobots (Hashemi-Petroodi et al., 2020). These resources can perform different tasks through a change in software. The flexible assembly line is becoming a necessity in today's manufacturing industry due to the increasing demand for mass customization, which requires the assembly of hundreds of items on the same line. Model variants may differ significantly, and when precedence graphs are incompatible, dynamic task assignment becomes necessary. The research studies we performed over the last few years extend model (3.1)-(3.11) to account for dynamic task assignment, and we consider three task assignment frameworks:

- *Model-dependent*: This framework assumes that a task o associated with two different items i and i' can be performed in different stations, but the task o of item i is always performed in the same station. We assume that a given task corresponds to the same process in different models, such as joining part A with part B. However, part A may differ in two different items i and i'. As a result, each task o requires the same equipment, but it may require more or less time for different items.

- *Planned movements*: In this framework, the assignment of tasks is decided at the operational level, but the equipment is fixed to the station. At the beginning of each period (e.g., a day), the process planner observes the production sequence, and he/she assigns the tasks to the stations.

- *Reactive assignment decisions*: The tasks are assigned in each takt to react to observation on the state of the line. Such observation may be related to the next item entering the line when the planner has no visibility on the sequence of incoming products. This observation may also be related to other sources of uncertainties such as variation in the process duration and resource breakdown.

The number of workers in each station is computed from the tasks assignment and takt time. As a result, the worker assignment strategy follows the task assignment framework.

While a few works have considered the dynamic assignment of workers in assembly line design (e.g., Battaïa et al., 2015), very few works have considered dynamic task assignment

(e.g., Kucukkoc and Zhang, 2014), and these works are restricted to the model-dependent decision framework.

3.2.1 Model-dependent task and workers assignment

The extension of model (3.1)-(3.11) to the model-dependent case requires modifying variable x_{so} into x_{soi}, where x_{soi} equals 1 if the task o of item i is performed in station s.

A major drawback of the resulting model (3.1)-(3.11) is that the number of production sequences can be very large. In (Hashemi-Petroodi et al., 2022), we propose defining the set of sequences Ω as all sequences that adhere to user-defined proportions u_i for each item i in each picture of the line, where a line picture refers to any subsequence of size $|\mathcal{S}|$. This constraint helps regulate the workload across the assembly line because it prevents an excessive number of complex items from being processed simultaneously. We show that there exists an efficient reformulation of the resulting model, and this model can handle cases with a large number of sequences in Ω. In model-dependent task and workers assignment, the number Y_{is} of workers at a station s depends only on the model i in the station. As a result, the objective function can be reformulated as:

$$\min \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} w_{se}c_e + f(Y_{11},\ldots,Y_{IS}), \tag{3.12}$$

where $f(Y_{11},\ldots,Y_{IS})$ represents the largest number of workers required for any pictures of the line that respect user-given ratio. The computation of Y_{is} does not require enumerating all scenarios. The value of $f(Y_{11},\ldots,Y_{IS})$ can be computed with the optimization problem:

$$f(Y_{11},\ldots,Y_{IS}) \quad = \max \quad \sum_{i\in\mathcal{I}} \sum_{s\in\mathcal{S}} f_{is}\ \alpha Y_{is} \tag{3.13}$$

s.t.

$$\sum_{s\in\mathcal{S}} f_{is} \quad \leq \quad u_i \qquad\qquad i \in \mathcal{I} \tag{3.14}$$

$$\sum_{i\in\mathcal{I}} f_{is} \quad = \quad 1 \qquad\qquad s \in \mathcal{S} \tag{3.15}$$

Model (3.13)-(3.15) assigns models variant to stations. The objective (3.13) is to maximize the number of workers, and the constraint ensures there is one model in each station and that

user-defined restrictions are respected. We show that the linear relaxation of (3.13)-(3.15) is an integer solution (Hashemi-Petroodi et al., 2022). As a result, the dualization per constraint commonly used in robust optimization (see section 2.4) applies. While the reformulation of the problem is relatively easy to solve using commercial solvers, the problem remains NP-hard. We propose a metaheuristic that successively solves the problem for a restricted number of stations. The local search iterates while a time limit is not reached. In each iteration, the local search selects a small number of successive stations, and it solves the problem only for these stations.

3.2.2 Planned tasks and workers assignment

In (Hashemi-Petroodi et al., 2024), we consider the *planned tasks and workers assignment* decision framework, where tasks can be reassigned to stations at the beginning of each period after observing the production sequence. Planned tasks and workers assignment requires modifying variable x_{so} into x^{ω}_{soi}, which equals 1 if the task o of the i^{th} item in the sequence performed in station s.

This leads to a two-stage robust problem, where the recourse problem is combinatorial. Not only does the application of the dualization per constraint not apply, but the application of the adversarial approach is also complex in this case. The classical approach for a two-stage robust optimization problem is to represent the recourse variables as affine functions of the unknown parameters. This approach does not work in our case because the recourse variables (affectation of task and worker) are binary variables.

Hashemi-Petroodi et al. (2024) formulate the case with planned task assignment as a scenario-based MILP. The model is similar to (3.13)-(3.15), but the task assignment x^{ω}_{soi} is scenario dependent for each task of each item. This scenario-based MILP does not scale well because the number of scenarios becomes too large for realistic size instances. In contrast to the model-dependent task assignment, the linear relaxation of the worst-sequence subproblem is not an integer solution in the planned task assignment decision framework. Therefore, we propose an adversarial approach to solve the problem.

We provide a generic description of the adversarial approach for robust optimization with combinatorial recourse actions, followed by its application in collaborative assembly lines. In Figure 3.1, we present the decomposition of the two-stage robust optimization approach with recourse into three subproblems. The first stage corresponds to the original scenario-based formulation, but the approach builds this set of scenarios iteratively. At each iteration, the solution of the first stage provides the values of the first-stage decisions, and

the second-step problem determines the worst-case scenario associated with the first-stage decisions. To find the worst-case scenario, the second-stage subproblem relies on the third-stage subproblem, which finds the optimal recourse action for each subproblem. The second- or third-stage subproblem may be an NP-hard combinatorial optimization problem. For the planned tasks and worker assignment application described below, we employ local search to solve the second- and third-stage subproblems. However, other approaches are also relevant. For instance, for the reactive case (see Section 3.2.3), we use an approximation that combines the second- and third-stage subproblems. In the application to generate robust machine learning examples in Section 4.3, we employ exact approaches to solve the second- and third-stage subproblems.

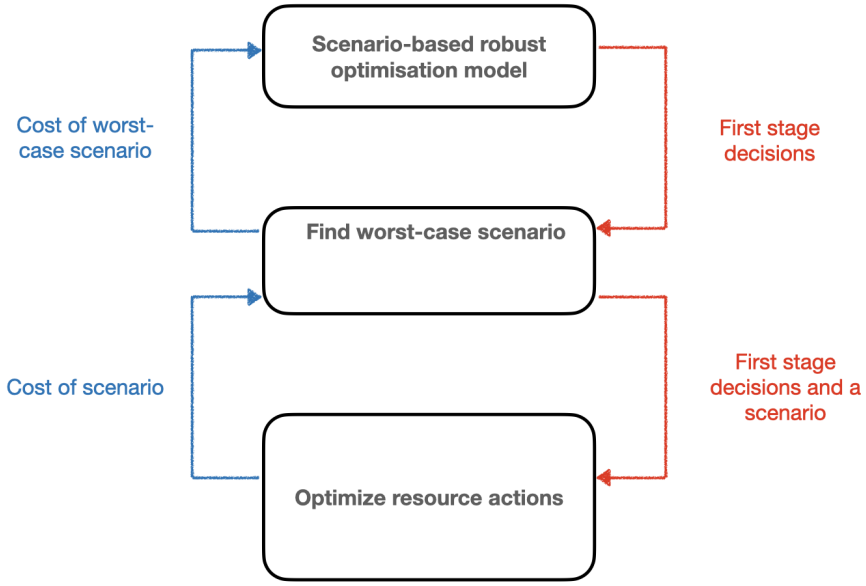


Figure 3.1: Three-step decomposition approach

Figure 3.2 shows the decomposition of the collaborative assembly line design problem with planned tasks and workers' assignments. The first stage assembly line design problem corresponds to scenario-based MILP, but with a subset only of the scenario. The result of this first step is a line design (assignment of equipment of station) and an assignment of tasks for each sequence that belongs to the current subset of scenarios. However, as the subset of sequences is not exhaustive, the first stage problem can underestimate the maximum number of workers required to process any incoming sequence of items. To estimate this number of

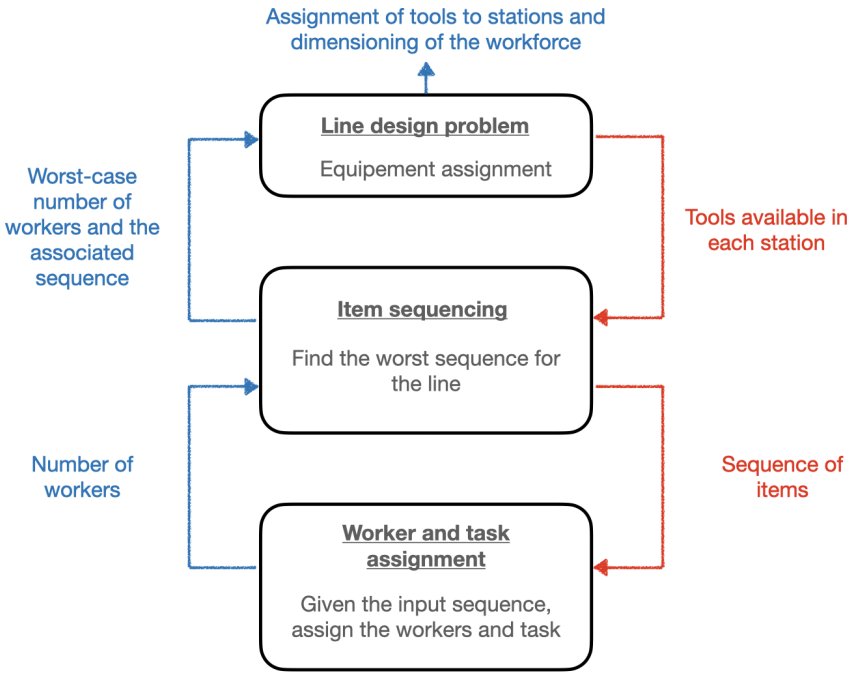


Figure 3.2: Steps of the heuristic for robust optimization with combinatorial subproblems

workers properly, the second-stage subproblem generates the sequence that maximizes the number of required workers for the current line design. We solve this sequencing problem using local search. The neighborhood of the current sequence $\hat{\omega}$ consists of solutions similar to $\hat{\omega}$, except for one model variant that is different from $\hat{\omega}$. To evaluate each sequence in the neighborhood, we solve an MILP that assigns tasks and workers. In other words, we seek to identify the worst sequence while optimizing task and worker assignments for a specific sequence. The output of the local search process is the worst sequence found and the associated number of required workers. Our results indicate that the approach is efficient because the local search finds near-optimal solutions, and the overall framework converges in a few iterations.

Comparing the heuristic solution approaches for robust optimization is not straightforward. Since the worst-case evaluation of a solution s relies on heuristics, there is no guarantee that the obtained costs are actual worst-case costs for solution s. As a result, a method m_1 may provide lower cost than a method m_2 not because the solution found by m_1 is better than the solution found by m_2, but because m_2 is better at finding the worst scenario. To ensure proper evaluation of the methods, we rely on simulations that randomly generate input sequences and evaluate all methods against the same set of random sequences.

Our simulation results show that the proposed approach works well even when the size of the sequences considered in optimization is not as large as they are in the simulation. In other words, a line designed by considering sufficiently large sequences will perform well even when the actual sequences are longer.

3.2.3 Reactive task and workers assignment

Our two recent works (Hashemi-Petroodi et al., 2023; Elyasi et al., 2024) study the scenarios where the tasks and workers are assigned in each takt after new information unfolds. Hashemi-Petroodi et al. (2023) follows our work on the design of collaborative lines that adapt to the incoming sequences of products. We consider the case where the planner has no visibility on the incoming sequence of items, and the decisions are made in each takt after observing the next item to enter the line. Elyasi et al. (2024) address the case where the process durations of tasks are unknown. After each takt, the planner observes the tasks completed at each station before assigning tasks and workers for the next takt.

Hashemi-Petroodi et al. (2023) formulate the worker and task assignment as an MDP. In this context, a state of the MDP describes the items in each station and the tasks already performed on each of these items. An action involves assigning tasks and workers to the stations. After an action, all items in the current sequence move to the next station, and the state can transition to $|K|$ different states corresponding to the arrival of each model variant at station 1. The transition probability reflects the probability of the first item in the arrival state. Figure 3.3 provides an example of a state, action, and transition. This figure displays the state with the model variant in each station and the graph of the remaining tasks to perform. Figure 3.3 shows possible transitions that correspond to the arrival of model variant 1, but other transitions exist.

We adopt an approach similar to the one presented in Section 2.5, where we convert the MDP into a linear program. The MDP is subject to constraints that correspond to decisions regarding the line design, such as the allocation of tools and the hiring of workers. The

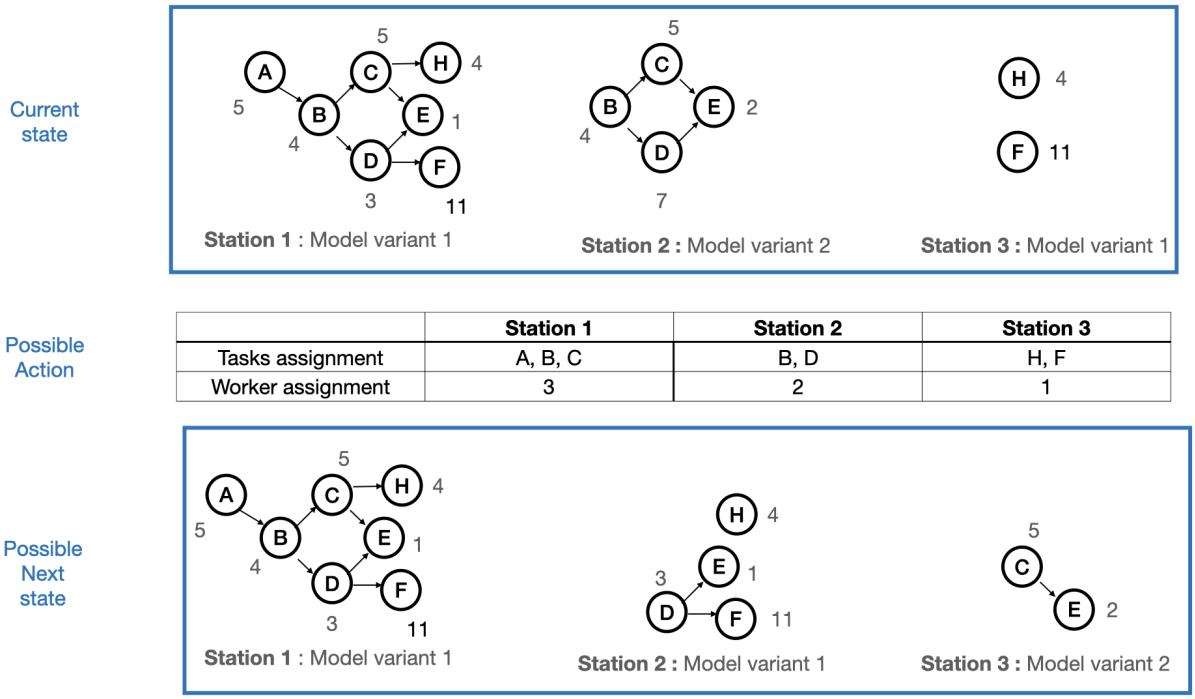


Figure 3.3: Example of a state action and transition in the MDP with sequence uncertainty

resulting MILP is formulated as follows:

$$\min \quad \alpha\, Y + \sum_{s\in\mathcal{S}} \sum_{e\in\mathcal{E}} W_{se} c_{se} \tag{3.16}$$

s.t.

$$\sum_{d\in\mathcal{D}} \sum_{a\in\mathcal{B}_d} Tr_a^{dd'} V_{ad} \;=\; \sum_{a'\in\mathcal{B}_{d'}} V_{a'd'} \qquad d'\in\mathcal{D} \tag{3.17}$$

$$\sum_{d\in\mathcal{D}} \sum_{a\in\mathcal{B}_d} V_{ad} \;=\; 1 \tag{3.18}$$

$$Z_a \geq V_{ad} \qquad a\in\mathcal{B}_d,\; d\in\mathcal{D} \tag{3.19}$$

$$Y \;\geq\; \sum_{s\in\mathcal{S}} q_s^a\, Z_a \qquad a\in\mathcal{B}_d,\; d\in\mathcal{D} \tag{3.20}$$

$$y_{soi}^a\, V_{ad} \;\leq\; \sum_{e\in E} r_{oe} W_{se} \qquad s\in\mathcal{S},\; o\in\mathcal{O},\; i\in\mathcal{I},\; a\in\mathcal{B}_d,\; d\in\mathcal{D} \tag{3.21}$$

$$W_{se}\in\{0,1\},\; 0\leq V_{ad}\leq 1 \tag{3.22}$$

$$Z_a\in\{0,1\},\; Y\geq 0 \tag{3.23}$$

Decision variables V_{ad} represent the probability of performing action a in state d, and

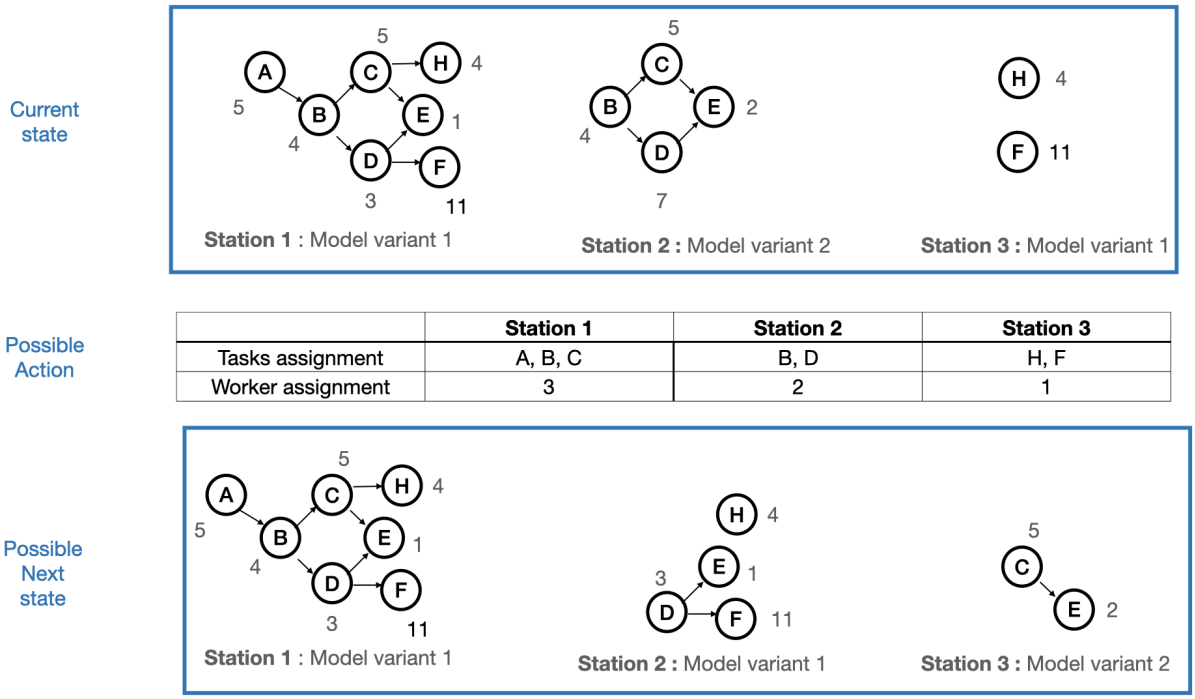
constraints (3.17)-(3.18) represent the standard constraints used to formulate infinite horizon MDPs as linear programs. Constraints (3.19) set the value of binary variables Z_a to 1 if the probability of selecting action a is greater than 0. Constraints (3.20) calculate the maximum number of required workers, where q_s^a denotes the number of workers in station s for action a. Constraints (3.21) prevent an action from being performed if the required equipment is not installed on the line. Here, y_{soi}^a is a parameter that equals 1 if action a requires processing task o of model i in station s. In addition to this robust formulation that minimizes the maximum number of workers required in any takt, Hashemi-Petroodi et al. (2023) also investigates the stochastic version, where the average number of workers is minimized.

Elyasi et al. (2024) focus on the case where the process duration is uncertain. The problem is formulated as a set of finite horizon MDPs, where each MDP corresponds to an item. The number of periods in the horizon corresponds to the number of stations. A state indicates the tasks that are yet to be performed when the item is in the considered station. An action indicates the task and worker assignment for a single station. The transition is computed based on tasks actually performed on the station. As the processing times are uncertain, only a subset of the selected tasks are performed, and the transition probability depends on the duration probability of each of the individual tasks. In this case, the ordering of the tasks in the station has an impact on the transition. Figure 3.4 shows possible transitions in the MDP that model process duration uncertainty in the same format as figure 3.3

Elyasi et al. (2024) translate the MDP of each item into linear programs coupled by the decisions on the design of the line. We propose a Benders decomposition approach where the master problem designs the line, and the subproblem solves the MDP of each item. The advantage of the Benders decomposition approach is that it maintains the combinatorial decisions associated with process planning (the affectation of tools) in the master problem. The decomposition allows handling larger-size MDPs in the subproblem because they are solved independently for each item.

3.3 Reconfigurable assembly lines

Nowadays, assembly lines are modified every 6 months to accomodate modifications of the product family. In other words, every 6 months, the company introduces a new generation of the product family, and the process engineers must reconfigure the line accordingly. Such modifications of the product family are necessary to meet the technological advancements and changes in supply, among other reasons. Manufacturers require assembly lines that can



	Station 1	Station 2	Station 3
Tasks assignment	A, B, C	B, D	H, F
Worker assignment	3	2	1

Figure 3.4: Example of a state, action, and transition in the MDP with uncertain processing time

be smoothly reconfigured over their life cycle (the next ten years) to adapt to changes in product requirements.

In (Mezghani et al., 2024), we consider the problem of designing a line while accounting for its reconfiguration throughout its life cycle. As the evolution of the product family is unknown, several scenarios that correspond to the sequence of product family generations are considered. The assembly line design problem requires assigning tasks and equipment to stations. The objective is to minimize the design and reconfiguration costs. As process engineers have no visibility into the evolution of the product family, the aim is to design a line that can be reconfigured at a reasonable cost for any path of product family generations that satisfies user-defined constraints. To avoid over-conservative solutions, user-defined constraints are imposed on the evolution of the joint precedence graph (Bryan et al., 2013) that represents a generation of the product family.

Our work is related to the research on reconfigurable manufacturing systems (RMS), which was introduced by Koren et al. (1999) to effectively deal with market changes, customized products, and volatile demand (Khettabi et al., 2021). One significant advantage of RMS is their ability to adapt to the changing product families in a cost-effective and

efficient manner (Koren et al., 1999). However, the design of an RMS that can reconfigure to handle all product variants in a family remains a challenging task (Altemeier et al., 2010). As optimization models for the design of manufacturing and assembly lines are similar, our work is also applicable in the context of RMS.

Given the proportion of each item of the family on average in the production sequence, the joint precedence graph is constructed by averaging the process duration of the tasks for each item. Figure 3.5 illustrates the construction of the joint precedence graph of items A and B, when item A represents 30% of the items in the sequence, and item B represents 70%. A line that respects the cycle time of the joint precedence graph provides the required throughput if there are buffers between stations to absorb differences in process duration. Therefore, the joint precedence graph approach is relevant when the line includes buffers between stations, and it allows using an algorithm developed for simple line balancing for the mixed model assembly line case.

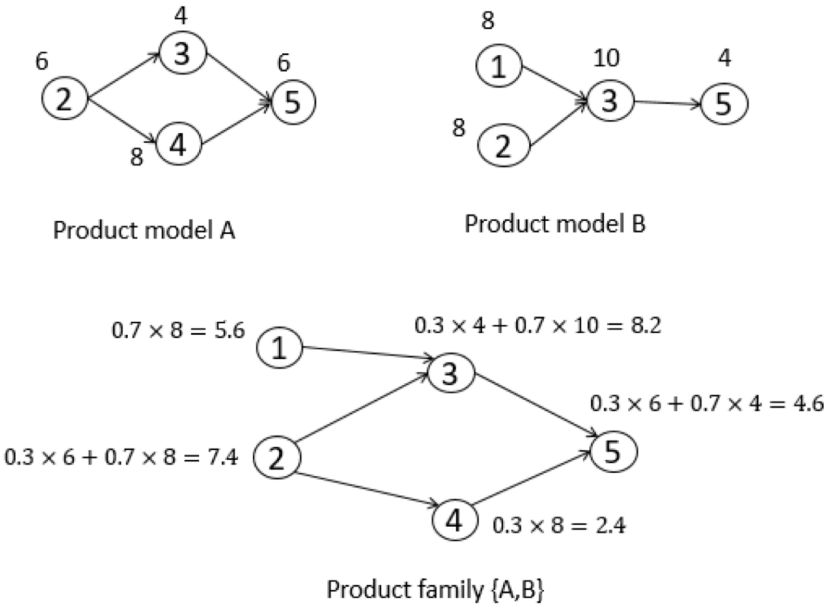


Figure 3.5: Joint precedent graph approach in MMAL

We suggests modifying the joint precedence graph in each generation (Mezghani et al., 2024). To control the proportion of changes between two generations, we define intervals for the number of tasks to add to the joint precedence graph, number of tasks to remove, and percentage of change in the process duration of the task. Given the current product

generation and its joint precedence graph, Mezghani et al. (2024) shows how to create the new product family from the new joint precedence graph. As the line is reconfigured every time a new generation of product family arrives, we model the decision process with a scenario tree as shown in Figure 3.6.

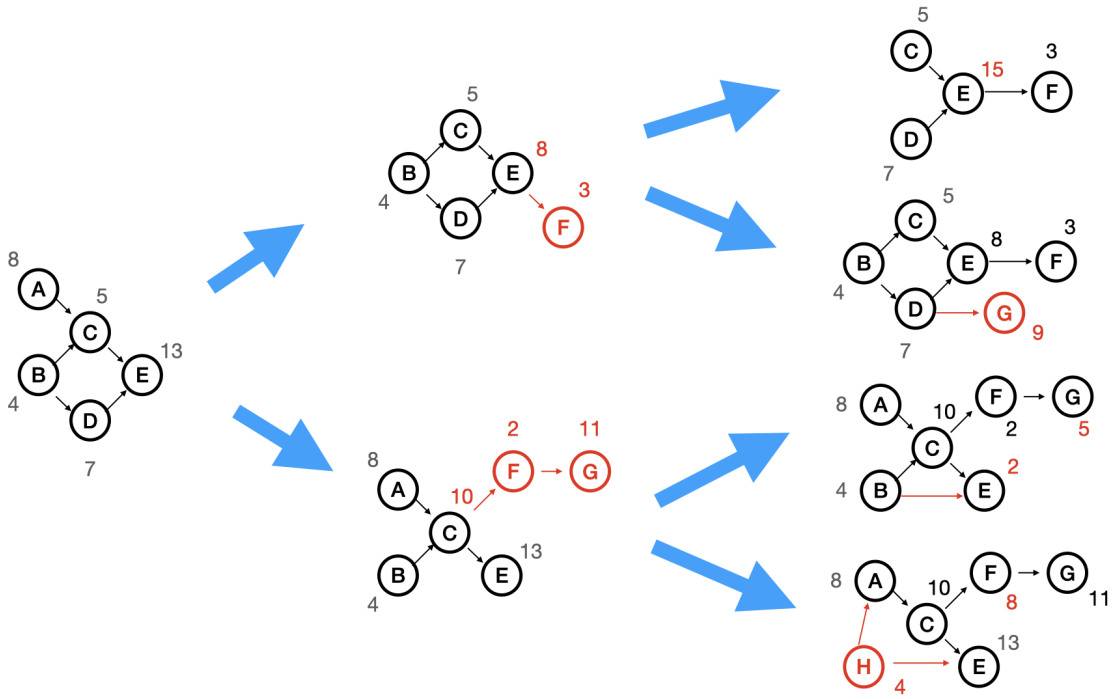


Figure 3.6: Scenario tree that represents the evolution of the product family

We consider a robust optimization version of the problem, where we optimize the line against the worst path of product family changes in the scenario tree. If the scenario tree includes all possible scenarios of the evolution of the production graph, the model provides the optimal solution. However, the approach is not practical because there are too many scenarios. The considered optimization problem is very challenging to solve because it is a multi-stage robust optimization problem where the recourse problem in each stage is an NP-Hard combinatorial optimization problem.

To solve the considered problem, we extend the approach proposed in (Hashemi-Petroodi et al., 2024) to the multi-stage decision problem. In the first stage, the problem solves the scenario tree formulation for a restricted number of scenarios. The second stage starts with the line design provided by the solution of the first stage, and it iteratively solves the problem of finding the worst joint precedence graph for the next generations. In this work, we suggest approximating the problem to a max-max to avoid the issue associated with the max-min

subproblems. In this approximation, the subproblem looks for the product family with the worst product reconfiguration, when the process planner designs the line to maximize the reconfiguration cost. We rely on the assumption that the resulting product family will be difficult to handle even when the process planner seeks to minimize the costs. Experimental results show the resulting approach scale well, and it provides solutions with costs close to the optimal solution on small instances.

We conducted experiments using a simulation of the proposed model. The results demonstrate that accounting for the entire life cycle of the assembly line leads to significant cost savings. Our approach provides around 16% cost savings when compared with that achieved using the classical method that re-optimizes the line with a deterministic model in each generation. Furthermore, our experimental results indicate that the proposed approach scales well and provides solutions with costs close to the optimal solutions in small instances.

3.4 Future works and perspectives

This section presents the perspective of research on assembly line design. We successively discuss the improvement of the models for flexible assembly lines to bring them closer to reality, consideration of model to deal with different types of uncertainties, and improvement of the methodology to solve large-scale instances.

3.4.1 Bring flexible lines design models closer to reality

Our initial studies on flexible assembly line can be extended in several ways to bring them closer to reality. This section succesively discusses the incorporation of human factors, the consideration of additional details (e.g., buffers between stations, detailed schedule at the station level, etc.), and the incorporation of sustainablity aspects.

Human factors in flexibles assembly lines design

In classical assembly lines, workers are fixed at the stations, and they perform the same set of tasks throughout the day. Unlike such static lines, our recent works (Hashemi-Petroodi et al., 2024, 2023) showed that dynamic tasks and moving workers yield more efficient assembly lines. The flexible assignment of tasks is now possible because of the availability of new resources such as programmable CNC machines or programmable robots/cobots. These resources can perform different tasks by a simple change on the software level. The flexible assembly line is also becoming a necessity with the arrival of mass customization that require

the assembly of hundreds of items on the same line. The model variants may vary significantly, and for instance, when precedence graphs are incompatible, dynamic task assignment becomes a necessity.

Although these previous works show that flexible assembly lines are efficient, this paradigm imposes a change on the way workers perform their activities. In such lines, workers are moving more, and they perform a wider set of tasks. The increase in the set of tasks may have negative effects on their productivity, and the loss of regularity may generate stress. On the positive side, their dynamicity may reduce boredom. Therefore, there is a need to analyze the precise impact of these flexible lines on the well-being of workers and their productivity, and to develop approaches for managing these side effects.

Flexible lines represent a chance to adapt the work process to workers. Workers have different profiles, and the objective is to reconfigure the line and assign tasks to workers such that the affectation matches the worker's profiles. Considering worker diversity increases workers' satisfaction and productivity (Battini et al., 2022). In this context, we must be able to evaluate the efficiency (process duration) and well-being (stress level vs. boredom) of workers when they are assigned a sequence of tasks. Thus, there is a need to develop tools to allocate workers to tasks in real time and to design the line such that it is sufficiently flexible to accommodate various teams of workers.

Finally, this research topic is in line with the industry 5.0 concept that requires placing humans at the center of the manufacturing systems. Further, the research on the ergonomic impact of dynamic task assignment does not fall within my area of expertise because it requires the knowledge of ergonomics. However, we already identified some partners to perform these investigations in a multi-disciplinary project.

Incorporating additional details in the models

The considered assembly line design models omit some details such as buffers between stations, movement of workers, and schedule of each worker on a station. These omissions are acceptable in initial studies because the problem becomes too complex to solve when these details are included. However, future works must evaluate the impact of omitting these details, and we must build on our experience to address the more complex models.

Worker movements pose an issue owing to the distance traveled by the workers in long assembly lines with short takt times. In this scenario, the travel time of workers is not negligible. Investigations on the incorporation of worker movements must study several variants of the model. For instance, a model could prevent workers from moving more than n stations

away from their current station, another model may constrain the number of workers moving in each takt, and a different could be used to segment the line such that workers stay within their segment. The worker movement strategy may also be studied. While we can consider a line where all workers move, we can also define a set of utility workers that move where needed, whereas other workers stay fixed at their station.

The studied models assume that the processing time of each task changes depending on the number of workers assigned to the station. In practice, workers work on separate tasks, and as tasks are often non preemptable, computing the precise time required to complete all tasks of the station requires solving a scheduling problem where the tasks are assigned to the workers. In this context, some authors (Sikora and Weckenborg, 2022) showed that Benders decomposition could perform well on related problems, and we can also exploit some properties of the scheduling subproblems.

Our models assume there are no buffers between stations, which is the case when assembling large items (e.g., cars, and plans, ...). However, for smaller items, buffers can be used to compensate for differences in the processing times between different product variants at each station. Owing to these buffers, the takt time can depend on the model. With the buffer, the takt time may change as needed, as long as the line can recover from the created delay. Additional studies are required to formulate the assembly line balancing problem with buffers and define an efficient solution approach for the resulting model.

Finally, we can study extensions of the model to consider the "and-or" precedence graph. Most studies on disassembly planning consider the "and-or" precedence graph (Li et al., 2020) that includes more details than those of the classical precedence graph considered in assembly line balancing. Although, there is no real reason to not consider the "and-or" graph in assembly planning, and future studies should evaluate the impact of using the and-or graph for assembly line design. The drawback is that the solution approaches that consider the "and-or" graph as input are more time-consuming.

Incorporate sustainability aspect in assembly line design

The work on reconfigurable assembly line initiated in (Mezghani et al., 2024) can be extended to account for sustainable aspects of line design. The proposed approach could be adapted to favor the circularity of production resources rather than their disposal. Machine and equipment selection could consider factors such as remaining lifespan and the ease of reselling to other companies. The model could also suggest renting equipment when its use is limited to a few production cycles.

Furthermore, the proposed approach could incorporate a workers' training plan to avoid hiring and firing workers when new qualifications are required. This would not only improve the sustainability of the line design but also help reduce labor costs and enhance worker satisfaction. Future studies could explore the feasibility of incorporating these sustainability considerations into the proposed approach and evaluate the impact of such considerations on the overall line design.

Other sources of uncertainties

Our works on the stochastic assembly line design problem focused on the uncertainty in the production sequence, process durations, and future generation of products. Other sources of uncertainty exist and should be studies, and they include employee absenteeism, machine breakdown, operation failure, ...

3.4.2 Extension to different decision frameworks

This section presents possbile extension of the models we have studied over the passed few years to accomodate different decision framework. First, we discuss improvement of the uncertainty set that describe the evolution of the product family. Second, we explain how the flexible line models can be extended to determine sequencing rules all incoming sequences must respect, rather than requiring the user to input these rules. Third, we discuss a modification of the MDP that deal with uncertain process duration to cope with a different assumption on the wway workers react when they cannot complete a task within the takt time.

Uncertainty description for product family evolution

The model that design a line that adapts smoothly to change in the product reconfiguration requires to properly define the possible evolution of the product family. In (Mezghani et al., 2024), we proposes a basic definition of the uncertainty set, where the different parameters (process duration, new tasks, etc.) evolve in the given intervals. Future work must investigate the definition of this uncertainty set. The solution of the proposed algorithm guarantees that the reconfiguration of the line is cost-efficient for any scenario in the uncertainty set. The uncertainty set must impose constraints on the evolution of the product family. These constraints must be easy to understand by process engineers and must remove unrealistic paths that would lead to over-conservative solutions. This investigation must be performed

with a simulation model that computes the impact of a given definiton of the uncertainty set, and the uncertainty set must be design in discussion with practitioners.

Additionally, our work (Mezghani et al., 2024) must be extended to consider the stochastic optimization version, where we minimize the expected cost over the possible product evolutions. Future work must investigate tools to estimate the probability distribution of parameters that describe the product's evolution. The resulting stochastic optimization with combinatorial resources is hard to solve, and we may investigate methods that rely on approximating the future reconfiguration cost of the line.

Selecting scheduling rules during the design of MMAL

Some of my works aim to design flexible assembly line design that can meet throughput requirements for various sequences of items entering the line. In these works, we assume that scheduling rules are given, and we optimize the line to accommodate the worst sequence that respects these rules. In practice, scheduling rules can be defined simultaneously while designing the line. These rules should strike a balance between being restrictive to remove sequences that prevent respecting the takt time and allowing flexibility to create efficient production schedules on the upstream resources.

For example, in car production (see https://www.roadef.org/challenge/2005/fr/sujet.php), the upstream workshop is a painting workshop, and the assembly sequence constraint must allow for efficient grouping of cars produced in the same color. To determine the impact of scheduling restriction on the performance of the upstream resource, we can calculate the number of sequences forbidden by a restriction, and the overall optimization approach may seek to minimize (or constrain) the number of removed sequences. Another approach may rely on scenarios that provide the number of models of each color, and the model selects the restrictions such that they do not remove good upstream schedules for the scenarios.

To properly design the assembly line with scheduling rules, additional studies are required to investigate these different approaches and identify the most efficient approach to define and incorporate the scheduling rules into the line design.

Consider uncertain task duration with variable takt time

Elyasi et al. (2024) consider assembly line design under process duration uncertainty, and we that assume the workers do not start a task if they are not sure if they can complete it on time. Extension of this work could consider a different strategy, where the workers complete the tasks even if it requires processing after the takt time. As a consequence, the takt time

must be reduced in the next station to ensure the work can be done.

3.4.3 Methods to deal with large-scale instances

Metaheuristics for robust optimization problem

Several stochastic variants of the assembly line design problem lead to scenarios where the recourse problem is combinatorial. For instance, (Hashemi-Petroodi et al., 2024) requires assigning tasks and workers to stations after observing the production sequence, and this recourse problem is NP-hard. Some situations (Mezghani et al., 2024) even lead to multi-stage problems with integer recourse.

To address these two cases, we proposed approaches that rely on heuristics to evaluate the cost for the worst-case scenario. A different approach could rely on the definition of rules to solve the recourse problems. Solving the subproblem that seeks the worst scenario becomes less challenging because the parameter of the rules becomes first-stage decisions. Although the affine rules commonly employed in adaptive robust optimization do not apply to the combinatorial problem, an avenue of research focuses on the design of more complex parameterized rules for these problems. Such rules may be based on common sense heuristics. For instance, a rule for worker and task assignment in (Hashemi-Petroodi et al., 2024) can rely on priority ranking for the tasks, and this priority list can be a first-stage decision.

Another type of policy may require solving an MILP. For instance, when designing a line that can be reconfigured throughout its life cycle (Mezghani et al., 2024), an MILP can suggest reconfiguration of the line in each generation. This MILP can include a reconfigurability indicator in the objective function, and this indicator may be an affine function of the characteristics of the line design. The parameter of this affine function may be the first-stage decision variable, and a multi-stage decision process can reduce it to a two-stage process where the second stage applies the MILP to a scenario path. Such an approach may be applicable to the stochastic variant of the problem.

The scenario tree generation approach proposed by Mezghani et al. (2024) deserves further investigation. The current method generates the scenarios in the tree node-by-node, and investigations are required to optimize the full scenario tree in one subproblem. In this context, the multi-stage process is divided into two steps. The first step designs the scenario tree and the second finds the solution that minimizes the worst-case costs on the scenario tree. The resulting method is an extension of the approach proposed by Hashemi-Petroodi et al. (2024) to multi-stage decision processes.

Metaheuristics for MDP-based line balancing

We presented two models where the assembly line design problem includes the optimization of an MDP. These two formulations are new, and our work shows their relevance. However, the solution approaches presented in these papers are limited to small-scale instances with few tasks. The design of approaches to deal with large-scale problems will b covered in future work.

There is a need to study policies for designing dynamic task and worker assignment policies. The current approaches are based on table look-up policies that specify the action to perform in each state. These policies do not scale because the number of states increases exponentially with the number of tasks. For a large number of tasks, the table will not fit in the memory. Other policies may be based on simple rules. For example, for the case with uncertain duration, a policy may include an ordering of the tasks to perform in the stations. Similarly, workers may be assigned based on the priority of the tasks. More complex policies may rely on linear programming or machine learning approaches (e.g., neural network policy). Thus, there is a need to propose different policies and evaluate their performance on a simulation.

Once we define an efficient policy, we will integrate it into heuristic and metaheuristics to design the line. One approach could be to use a local search, where the neighborhood contains tool assignments slightly different from the current line design, and the line's design evaluation requires simulating the policy on incoming sequence samples. In this context, evaluating the solution will be time-consuming, and we may investigate the use of machine learning to filter non interesting moves. The initial solution may come from a greedy heuristic, or it can be the solution of the static version of the problem.

Chapter 4

Mathematical programs as models in digital twins

This section describes the vision of the intelligent digital twin that considers mathematical models as digital twins (section 4.1). The realization of this intelligent digital twin concept requires collaborative research, and we briefly describe the work performed in the collaborative research project ASSISTANT in section 4.2. The realization of the intelligent digital twin concept requires two research directions in my field of expertise. First, the mathematical model must be a precise representation of the workshop, and section 4.3 discusses our work related to the learning of constraints in mathematical programming. We combine machine learning models with mathematical programs to automatically increase the precision of the mathematical model from data. Second, as the model will never be perfect, the mathematical model must account for uncertainty, and our contributions to optimization under uncertainties are already mentioned in sections 2 and 3.

4.1 Intelligent digital twin concept

There exist various definitions of a digital twin in the literature (Negri et al., 2017). However, Glaessgen and Stargel (2012) provide the following generally accepted definition in the context of aero-space safety: "*A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin. The Digital Twin is ultra-realistic and may consider one or more important and interdependent vehicle systems, including airframe, propulsion and energy storage, life*

support, avionics, thermal protection, etc. "

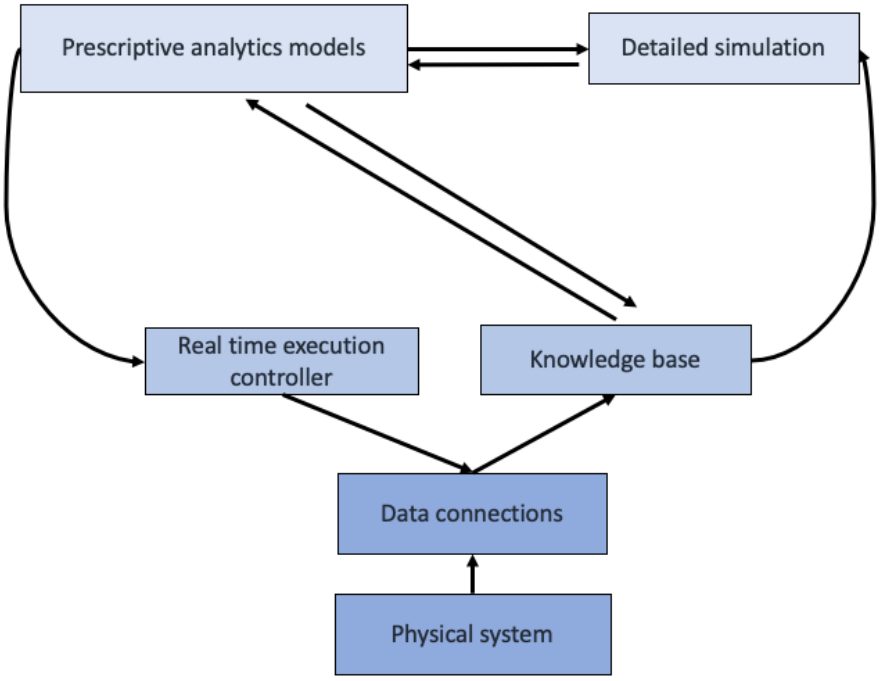
A classical digital twin requires five elements: (1) a physical object, (2) a virtual model, (3) data, (4) data connections, and (5) services provided to the end users. In addition, a digital twin provides the following characteristics:

1. Data are collected from the physical object and sent to the model automatically.
2. The computer model stays in synchronization with a physical object. Any change in the physical object must be passed on to the virtual models.
3. The model can pass instructions to the physical object.
4. The model accounts for uncertainties. The model must account for uncertainties in the environment of the physical object because it includes some parameters that can never be forecasted perfectly. Any model is only an approximation of the complex real world. The model should be sufficiently robust to provide valid decisions despite these approximations.

This definition is sufficiently broad to encompass any type of physical object, and any type of virtual model (simulation, mathematical models, data model, etc.). This might be an explanation for the rising interest among researchers and in the industry on the digital twin topic. The digital twin is becoming a core concept of the industry 4.0 revolution because it is a broad concept that can gather all technologies used in computer science for manufacturing.

We define the intelligent digital twin concept as the application of the digital twin in the context of prescriptive analytic models. Prescriptive analytics relies on optimization models that represent the production system through mathematical equations. Classical prescriptive models are designed for specific applications, and any change in the manufacturing process requires a change in the model. In addition, these models (like any model) are approximations of reality, and the classical implementations do not account for the uncertainty in the model.

An intelligent digital twin is an optimization model that can learn some of its parameters and constraints automatically to enhance its accuracy and remain synchronized with the shop floor. Such tools integrate machine learning techniques with optimization. On the one hand, AI methods can be used to learn the parameters of the models efficiently. On the other hand, AI methods can acquire models automatically to learn the functioning of constantly changing production systems. The intelligent digital twin also relies on optimization under uncertainty approaches to make reliable decisions. Figure 4.1 presents the main elements of



htp]

Figure 4.1: Main elements of the intelligent digital twin

an intelligent digital twin. Besides the prescriptive analytic module, the intelligent digital twin includes:

1. *Data collection and management*: Data flow from the data connector into a knowledge graph that eases the integration of data from several sources. Thus, the knowledge graph contains cleaned, up-to-date, and organized data. The domain model presents a rich data model that can be understood by the end user.

2. *Simulation*: The learning process of the prescriptive analytics software requires data, and these data are often generated by trial and error. A detailed simulation of the manufacturing system is necessary to generate these examples.

3. *Real-time execution*: Once decisions are made, they must be actuated on the shop floor. The real-time execution controller ensures the shop floor follows the process, and it triggers re-scheduling in case of deviation.

4.2 ASSISTANT project

The realization of the intelligent digital twin concept requires collaborative research. To realize this vision, the ASSISTANT project gathers 12 institutions with key experts in knowledge graphs, simulation, optimization, constraint acquisition, and AI ethics. I was the scientific coordinator of this project. Figure 4.2 shows the global picture of the software developed in the ASSISTANT project. Note that each box requires the use of various software. The ASSISTANT project focuses on the manufacturing industry, and we aim to develop four intelligent digital twins corresponding to different decision levels, namely, process planning, production planning, scheduling, and real-time execution. To complement the decision support systems, a data fabric ensures the collection of data from software and the shop floor. The data fabric is the main data provider of the tools, and it ensures communication with the shop floor and between the tool. Depending on the need of the tools, the data fabric provides raw data in real time, or clean, reliable, and organized data. Finally, the ASSISTANT project considers issues related to ethics in the development of software for the manufacturing industry. To ensure that humans remain responsible for all decisions, we offer a graphical interface to validate decisions made by the AI system or to control the real-time control policies.

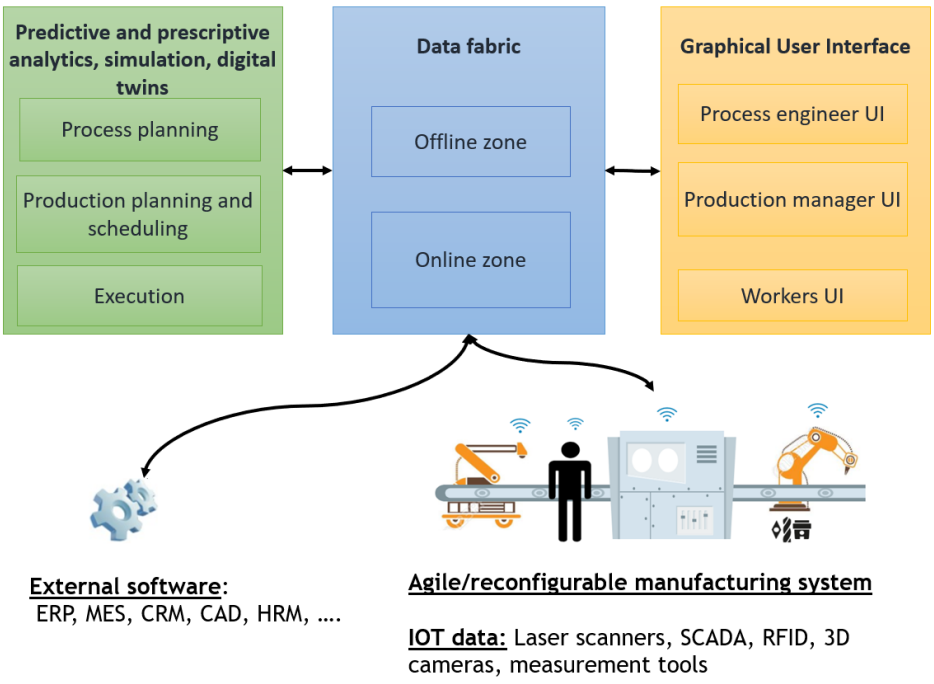


Figure 4.2: Main elements of the intelligent digital twin

4.3 Learning constraints automatically

We have studied the concept of learning part of the optimization model in the Thesis of David Tremblet. In this thesis we focus on learning the capacity constraint in the LSP. In a manufacturing environment, the output of the lot-sizing model provides a production plan, and the output of the production plan is the input for production scheduling. In each period of the horizon, the production plan indicates the quantity of each item to produce. Each of these production loads is translated into a set of jobs to be scheduled on the machines on the shop floor. The lot-sizing model includes a capacity constraint that ensures the production plan is feasible. However, the classical constraint is a rough approximation of the capacity constraint, and in practice, the output of the production planner is often infeasible.

In a recent work (Tremblet et al., 2024b), we investigated automatic constraint learning to better represent the production capacity during production planning. Classical approaches (Voss and Woodruff, 2006) rely on simple formulas to compare the load to the capacity, and they cannot account for the complex production processes encountered in today's shop floor (as resources can perform a wider range of tasks). We investigate how machine learning can help automatically learn the production capacity from the data. In (Tremblet et al., 2024a), we study various machine learning algorithms to predict the makespan of a scheduling problem. Our results suggest that machine learning models provide acceptable makespan prediction. These approximations are suitable for predicting capacity consumption. We show that approximations relying on a regression tree or neural network provide better accuracy (the makespan is closer to its optimal value) than applying common dispatching rules. The result is interesting when negotiating due dates or order acceptance with customers. The planner sees in real time if the factory has sufficient capacity to handle the customer's request.

In (Tremblet et al., 2024b), we investigate the translation of the machine learning model with a set of variables and constraints to directly incorporate the estimated capacity consumption into the mathematical model. The translation of linear regression is straightforward, and there exists approaches to translate other models such as decision trees (Biggs and Hariss, 2017), neural network (Fischetti and Jo, 2018), among others. Our results show that incorporating a machine learning model directly into an optimization model leads to poor results. Minimizing the objective function often yield solution where the plan consumes all the available capacity, and the optimizer often exploits the constraint approximation errors to decrease the costs. To circumvent these shortcomings, we propose (Tremblet et al., 2024b) constraining the training of the machine learning model such that the approximation does

not underestimate the capacity consumption in the training sample. In addition, we propose an adversarial training approach. The approach iteratively generates training examples that are not well predicted by the current approximation until no such example exists.

Generating adversarial training samples requires solving a max-min problem. The objective is to find lot sizes that maximize prediction error, where computing the actual makespan requires solving a minimization problem. To solve this problem, we decompose it into three subproblems, as shown in Figure 4.3. The first stage fits the approximation with the current training examples. The second-stage subproblem generates a training sample where the capacity consumption is underestimated by the current approximation. This second model includes a set of conjunctive graphs that define solutions to production schedules, and these graphs help us compute the makespan associated with a production plan. There exist a large number of conjunctive graphs, and we generate them iteratively with the third subproblem. The third subproblem generates a conjunctive graph that yields the minimum makespan for the current scheduling problem.

There exist a large number of conjunctive graphs, and therefore, the exact adversarial training example generation approach is time-consuming. We propose solving the second subproblem using a heuristic approach that randomly samples lot sizes with resource consumption close to the capacity. We stop generating samples once 1000 samples are drawn without underestimating the capacity. In summary, the max-min problem is decomposed into three subproblems, and the second subproblem is solved using a heuristic approach due to the computational complexity of generating exact adversarial training examples.

4.4 Future works

The concept of digital twins is gaining popularity in the manufacturing industry because it integrates many Industry 4.0 technologies. I believe that operations research approaches are crucial for decision making in the manufacturing industry, and we must continue to push for integrating these tools into digital twins. Complex challenges in the development of digital twins include automatic data collection, data cleaning, reconciliation between data from different sources, and safe actuation. The solutions to these challenges do not fall within my area of expertise. However, improving the precision of mathematical models with data is an interesting avenue for future research. David Tremblet's thesis investigates the approximation of the capacity constraint in the lot-sizing model, and this approach could apply to other models. For instance, in assembly line design, the precedence constraint does not depend only

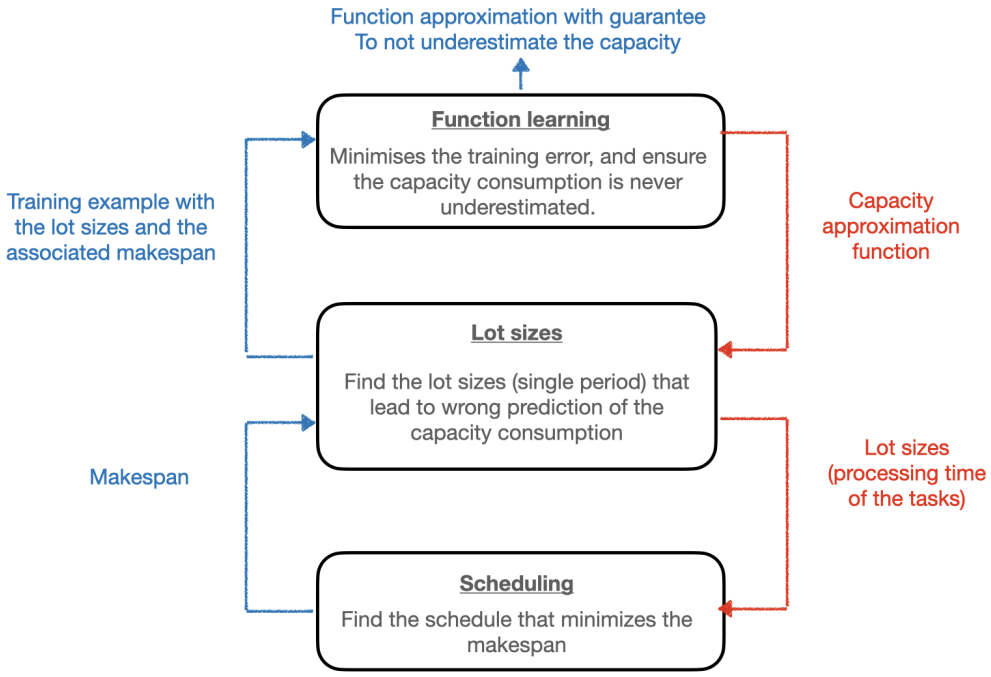


Figure 4.3: Decomposition of the adversarial training approach in three subproblems.

on the sequence of operations but also on the used tools. However, it is considerably simpler to approximate these constraints with a simple precedence graph. Therefore, predicting the feasibility of an assembly plan with machine learning is another interesting research direction.

The first chapter of David Tremblet's thesis is an initial work conducted in a controlled environment, where we checked if the plans are feasible by solving a scheduling problem. Future work must investigate the possibility of learning capacity consumption from real data collected from an MES. An intermediate step might be to study the case where feasibility on the shop floor is checked in a detailed simulation. Such a detailed simulation will provide data for complex shop floors with many machines and jobs, and it may help incorporate the instability commonly encountered in workshops, where a given production load may be feasible in a week but not in the next one (because of machine breakdown or other uncertainties). Other interesting avenues for future research include the generalization of machine learning tools to multi-level LSPs, as well as the consideration of other machine learning models such as neural networks.

Chapter 5

Conclusion

The research I conducted after my PhD focused on the solution approach for lot sizing and assembly line design under uncertainty. These research studies are timely because supply chains are facing challenges, and the manufacturing industry needs robust and resilience manufacturing systems and supply chains. The proposed approaches provide robustness because the models account for various sources of uncertainties. The considered methods also improve resilience because the model accounts for the dynamic of the decision processes. In addition, the manufacturing industry is undergoing the industry 4.0 resolution that implies a digitalization of the factory. As a result, an increasing amount of data is being collected from the shopfloor and from various software. These data are useful for building the distribution or uncertainty sets input to optimization under uncertainty approaches.

A major challenge is the computational burden because the optimization approach under uncertainty often consumes a large amount of CPU time and memory. This computational burden is common in combinatorial optimization, and accounting for uncertainty increases the difficulty. Besides the combinatorial explosion of the number of decision variables, the number of scenarios in an uncertain environment grows exponentially with the number of periods. In a dynamic decision context, we attach recourse problems to each scenario, which further increases the difficulty. Therefore, our works focused on designing an efficient optimization approach. They include exact methods based on decomposition, approximation with scenario sampling, and various heuristics.

Over the past few years, I gained experience in the management of research projects and young researchers. While my PhD, postdoc, and industrial experience gave me the skills required to conduct research on my own and manage my activities, they did not prepare me to manage others. I had the chance to rely on experienced researchers from the MODELIS

team that helped me deal with various situations, and these situations could have been stressful without their advice. While these research management activities require different skills than conducting research, they are also very enjoyable. I particularly like to discuss research with PhD and postdoc students.

Over the past few years, I worked on various projects where I could transfer some of the research results to the industry. Further, I was able to validate the research result with manufacturing data. I had the chance to link my research and my teaching activities. I developed a teaching module on optimization under uncertainty, and several student's projects, examples, and exercises in the teaching module come directly from my research activities. These links between teaching and research also arise with various Master's projects I supervised.

Section 2.6, 3.4, and 4.4 provide several perspectives for future research. I will only recall the three main avenues of research in this chapter. First, we must investigate metaheuristics for optimization under uncertainty in applications with combinatorial recourse problems where the exact method performs poorly. Second, there is a need to integrate data analysis and predictive techniques in our investigation of optimization methods. Third, many of the models developed in the past year require further investigation to incorporate details and constraints encountered in practice.

Several ongoing research projects will focus on these perspectives over the next few years. In particular, the research project ALICIA started in January 2023 to investigate the design of reconfigurable and sustainable assembly lines. This project will fund two PhDs and a postdoc to carry out some parts of the future works mentioned in Section 3.4. In addition, the thesis of Haed Tavaokoly in collaboration with Oncu Hazir and Maher AGi in Rennes Schools of Business will extend our works on collaborative assembly lines. In the coming years, I also plan to continue working on approaches that link operations research with machine learning or reinforcement learning. For instance, this is the topic of the PhD thesis of Joseph Tompson that started in January 2023.

Bibliography

A. Agra, M. Poss, and M. Santos. Optimizing make-to-stock policies through a robust lot-sizing model. *International Journal of Production Economics*, 200:302–310, 2018.

D. Alem, E. Curcio, P. Amorim, and B. Almada-Lobo. A computational study of the general lot-sizing and scheduling model under demand uncertainty via robust and stochastic approaches. *Computers & Operations Research*, 90:125–141, 2018.

M. A. Aloulou, A. Dolgui, and M. Y. Kovalyov. A bibliography of non-deterministic lot-sizing models. *International Journal of Production Research*, 52(8):2293–2310, 2014.

S. Altemeier, M. Helmdach, A. Koberstein, and W. Dangelmaier. Reconfiguration of assembly lines under the influence of high product variety in the automotive industry–a decision support system. *International Journal of Production Research*, 48(21):6235–6256, 2010.

E. Asgari, S. Thevenin, and M. Mohammadi. Joint pricing, inventory, and lead time decisions in an ato system, 2024.

O. Battaïa, X. Delorme, A. Dolgui, J. Hagemann, A. Horlemann, S. Kovalev, and S. Malyutin. Workforce minimization for a mixed-model assembly line in the automotive industry. *International Journal of Production Economics*, 170:489–500, 2015.

D. Battini, N. Berti, S. Finco, I. Zennaro, and A. Das. Towards industry 5.0: A multi-objective job rotation model for an inclusive workforce. *International Journal of Production Economics*, 250:108619, 2022.

C. Becker and A. Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3):694–715, 2006.

N. Beldiceanu, M. I. Restrepo, and H. Simonis. Parameterised bounds on the sum of variables in time-series constraints. In *Integration of Constraint Programming, Artificial Intel-*

ligence, and Operations Research: 17th International Conference, CPAIOR 2020, Vienna, Austria, September 21–24, 2020, Proceedings, pages 82–98. Springer, 2020.

P. Beraldi, G. Ghiani, A. Grieco, and E. Guerriero. Fix and relax heuristic for a stochastic lot-sizing problem. *Computational Optimization and Applications*, 33(2-3):303, 2006.

D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations research*, 54(1):150–168, 2006.

D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5 (2):389–414, 2008.

M. Biggs and R. Hariss. Optimizing objective functions determined from random forests. *SSRN Electronic Journal*, 2017.

R. Blanquero, E. Carrizosa, C. Molero-Río, and D. R. Morales. On optimal regression trees to detect critical intervals for multivariate functional data. *Computers & Operations Research*, page 106152, 2023.

P. Bonami, A. Lodi, and G. Zarpellon. A classifier to decide on the linearization of mixed-integer quadratic problems in cplex. *Operations Research*, 70(6):3303–3320, 2022.

N. Boysen, M. Fliedner, and A. Scholl. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111(2):509–528, 2008.

P. Brandimarte. Multi-item capacitated lot-sizing with demand uncertainty. *International Journal of Production Research*, 44(15):2997–3022, 2006.

A. Bryan, S. J. Hu, and Y. Koren. Assembly system reconfiguration planning. *Journal of Manufacturing Science and Engineering*, 135(4), 2013.

Q. Cappart, T. Moisan, L.-M. Rousseau, I. Prémont-Schwarz, and A. A. Cire. Combining reinforcement learning and constraint programming for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3677–3687, 2021.

E. Carrizosa, V. Guerrero, and D. Romero Morales. On mathematical optimization for clustering categories in contingency tables. *Advances in Data Analysis and Classification*, pages 1–23, 2022.

G. Castañé, A. Dolgui, N. Kousi, B. Meyers, S. Thevenin, E. Vyhmeister, and P.-O. Östberg. The assistant project: AI for high level decisions in manufacturing. *International Journal of Production Research*, pages 1–19, 2022.

X. Delorme, A. Dolgui, S. Kovalev, and M. Y. Kovalyov. Minimizing the number of workers in a paced mixed-model assembly line. *European Journal of Operational Research*, 272(1): 188–194, 2019.

M. Elyasi, S. Thevenin, A. Cerqueus, and A. Dolgui. Mixed-model assembly lines with process time uncertainty, 2024.

M. Fischetti and J. Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, 2018. ISSN 1383-7133.

E. Glaessgen and D. Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, page 1818, 2012.

C. Gonnermann, S. E. Hashemi-Petroodi, S. Thevenin, A. Dolgui, and R. Daub. A skill- and feature-based approach to planning process monitoring in assembly planning. *The International Journal of Advanced Manufacturing Technology*, 122(5-6):2645–2670, 2022.

S. E. Hashemi-Petroodi, S. Thevenin, S. Kovalev, and A. Dolgui. Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. *Annual Reviews in Control*, 49:264–276, 2020.

S. E. Hashemi-Petroodi, A. Dolgui, S. Kovalev, M. Y. Kovalyov, and S. Thevenin. Workforce reconfiguration strategies in manufacturing systems: a state of the art. *International Journal of Production Research*, 59(22):6721–6744, 2021.

S. E. Hashemi-Petroodi, S. Thevenin, S. Kovalev, and A. Dolgui. Model-dependent task assignment in multi-manned mixed-model assembly lines with walking workers. *Omega*, 113:102688, 2022.

S. E. Hashemi-Petroodi, S. Thevenin, S. Kovalev, and A. Dolgui. Markov decision process for multi-manned mixed-model assembly lines with walking workers. *International Journal of Production Economics*, 255:108661, 2023.

S. E. Hashemi-Petroodi, S. Thevenin, S. Kovalev, and A. Dolgui. Adversarial approach for the robust balancing of mixed-model assembly lines with dynamic task assignment, 2024.

Ö. Hazır and A. Dolgui. Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering*, 65(2):261–267, 2013.

S. Helber, F. Sahling, and K. Schimmelpfeng. Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum*, 35:75–105, 2013.

T. Homem-de Mello and G. Bayraksan. Stochastic constraints and variance reduction techniques. *Handbook of simulation optimization*, pages 245–276, 2015.

R. Jans and Z. Degraeve. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, 46(6):1619–1643, 2008.

I. Khettabi, L. Benyoucef, and M. A. Boutiche. Sustainable reconfigurable manufacturing system design using adapted multi-objective evolutionary-based approaches. *The International Journal of Advanced Manufacturing Technology*, 115(11):3741–3759, 2021.

Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *Annals of the CIRP*, 48:2, 1999.

I. Kucukkoc and D. Z. Zhang. Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *International Journal of Production Research*, 52(12): 3665–3687, 2014.

P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46 (9):1214–1235, 2000.

J. Li, D. Pang, Y. Zheng, X. Guan, and X. Le. A flexible manufacturing assembly system with deep reinforcement learning. *Control Engineering Practice*, 118:104957, 2022.

Z. Li, Z. A. Çil, S. Mete, and I. Kucukkoc. A fast branch, bound and remember algorithm for disassembly line balancing problem. *International Journal of Production Research*, 58 (11):3220–3234, 2020.

D. Liu, V. Perreault, A. Hertz, and A. Lodi. A machine learning framework for neighbor generation in metaheuristic search. *arXiv preprint arXiv:2212.11451*, 2022.

M.-A. Louly, A. Dolgui, and F. Hnaien. Supply planning for single-level assembly system with stochastic component delivery times and service-level constraint. *International Journal of Production Economics*, 115(1):236–247, 2008.

D. Luo, S. Thevenin, and A. Dolgui. A state-of-the-art on production planning in industry 4.0. *International Journal of Production Research*, pages 1–31, 2022.

D. Luo, S. Thevenin, and A. Dolgui. Fix and optimize heuristic for stochastic lot-zizinf, 2024.

N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021.

P. Metzker, S. Thevenin, Y. Adulyasak, and A. Dolgui. Robust optimization for lot-sizing problems under yield uncertainty. *Computers & Operations Research*, 149:106025, 2023a.

P. Metzker, S. Thevenin, Y. Adulyasak, and A. Dolgui. Adaptive robust optimization for lot-sizing under yield uncertainty, 2023b.

P. Metzker, S. Thevenin, Y. Adulyasak, and A. Dolgui. Distributionally robust optimization for the multi-period multi-item lot-sizing problems under yield uncertainty, 2024.

Y. Mezghani, S. E. Hashemi-Petroodi, S. Thevenin, and A. Dolgui. Design and reconfiguration planning of mixed-model assembly lines under uncertain evolutions of product family, 2024.

G. Michalos, N. Kousi, P. Karagiannis, C. Gkournelos, K. Dimoulas, S. Koukas, K. Mparis, A. Papavasileiou, and S. Makris. Seamless human robot collaborative assembly–an automotive case study. *Mechatronics*, 55:194–211, 2018.

M. C. O. Moreira, J.-F. Cordeau, A. M. Costa, and G. Laporte. Robust assembly line balancing with heterogeneous workers. *Computers & Industrial Engineering*, 88:254–263, 2015.

E. Negri, L. Fumagalli, and M. Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

N. Niroumandrad, N. Lahrichi, and A. Lodi. *Learning Tabu Search Algorithms: A Scheduling Application*. Bureau de Montreal, Université de Montreal, 2022.

E. Özceylan, C. B. Kalayci, A. Güngör, and S. M. Gupta. Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, 57(15-16):4805–4827, 2019.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

F. Quezada, C. Gicquel, and S. Kedad-Sidhoum. Combining polyhedral approaches and stochastic dual dynamic integer programming for solving the uncapacitated lot-sizing problem under uncertainty. *INFORMS Journal on Computing*, 34(2):1024–1041, 2022.

B. Rekiek, A. Dolgui, A. Delchambre, and A. Bratcu. State of art of optimization methods for assembly line design. *Annual Reviews in Control*, 26(2):163–174, 2002.

K. W. Ross and R. Varadarajan. Multichain markov decision processes with a sample path constraint: A decomposition approach. *Mathematics of Operations Research*, 16(1):195–207, 1991.

C. G. S. Sikora and C. Weckenborg. Balancing of assembly lines with collaborative robots: comparing approaches of the benders' decomposition algorithm. *International Journal of Production Research*, pages 1–17, 2022.

E. A. Silver, D. F. Pyke, R. Peterson, et al. *Inventory management and production planning and scheduling*, volume 3. Wiley New York, 1998.

I. Slama, O. Ben-Ammar, A. Dolgui, and F. Masmoudi. A stochastic model for a two-level disassembly lot-sizing problem under random lead time. In *Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems: IFIP WG 5.7 International Conference, APMS 2020, Novi Sad, Serbia, August 30–September 3, 2020, Proceedings, Part I*, pages 275–283. Springer, 2020.

I. Slama, O. Ben-Ammar, S. Thevenin, A. Dolgui, and F. Masmoudi. Stochastic program for disassembly lot-sizing under uncertain component refurbishing lead times. *European Journal of Operational Research*, 303(3):1183–1198, 2022.

I. Slama, O. Ben-Ammar, S. Thevenin, A. Dolgui, and F. Masmoudi. A stochastic model for a multi-echelon disassembly lot-sizing problem under random lead times., 2024.

D. Tavaghof-Gigloo and S. Minner. Planning approaches for stochastic capacitated lot-sizing with service level constraints. *International Journal of Production Research*, pages 1–21, 2020.

H. Tempelmeier. Stochastic lot sizing problems. *Handbook of stochastic models and analysis of manufacturing system operations*, pages 313–344, 2013.

S. Thevenin and Y. Adulyasak. Scalable multi-item inventory planning for slow-movers: A mathematical programming approach, 2024.

S. Thevenin and N. Zufferey. Learning variable neighborhood search for a scheduling problem with time windows and rejections. *Discrete Applied Mathematics*, 261:344–353, 2019.

S. Thevenin, N. Zufferey, and M. Widmer. Metaheuristics for a scheduling problem with rejection and tardiness penalties. *Journal of Scheduling*, 18(1):89–105, 2015.

S. Thevenin, N. Zufferey, and M. Widmer. Order acceptance and scheduling with earliness and tardiness penalties. *Journal of Heuristics*, 22:849–890, 2016.

S. Thevenin, N. Zufferey, and R. Glardon. Model and metaheuristics for a scheduling problem integrating procurement, sale and distribution decisions. *Annals of Operations Research*, 259(1):437–460, 2017a.

S. Thevenin, N. Zufferey, and J.-Y. Potvin. Makespan minimisation for a parallel machine scheduling problem with preemption and job incompatibility. *International Journal of Production Research*, 55(6):1588–1606, 2017b.

S. Thevenin, N. Zufferey, and J.-Y. Potvin. Graph multi-coloring for a job scheduling application. *Discrete Applied Mathematics*, 234:218–235, 2018.

S. Thevenin, Y. Adulyasak, and J.-F. Cordeau. Material requirements planning under demand uncertainty using stochastic optimization. *Production and Operations Management*, 30(2):475–493, 2021.

S. Thevenin, Y. Adulyasak, and J.-F. Cordeau. Stochastic dual dynamic programming for multiechelon lot sizing with component substitution. *INFORMS Journal on Computing*, 34(6):3151–3169, 2022a.

S. Thevenin, O. Ben-Ammar, and N. Brahimi. Robust optimization approaches for purchase planning with supplier selection under lead time uncertainty. *European Journal of Operational Research*, 303(3):1199–1215, 2022b.

S. Thevenin, N. Mebarki, and P. Chatellier. Dynamic scheduling of a picking robot with limited buffer and rejection: an industrial case study. *International Transactions in Operational Research*, 29(3):1394–1416, 2022c.

C. F. M. Toledo, M. da Silva Arantes, M. Y. B. Hossomi, P. M. França, and K. Akartunalı. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. *Journal of Heuristics*, 21:687–717, 2015.

D. Tremblet, S. Thevenin, and A. Dolgui. Makespan estimation in a flexible job-shop scheduling environment using machine learning, 2024a.

D. Tremblet, S. Thevenin, and A. Dolgui. Learning the capacity consumption of lot-sizing models from production schedules, 2024b.

E. Tuncel, A. Zeid, and S. Kamarthi. Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning. *Journal of Intelligent Manufacturing*, 25: 647–659, 2014.

S. Voss and D. L. Woodruff. *Introduction to computational optimization models for production planning in a supply chain*, volume 240. Springer Science & Business Media, 2006.

G. Zarpellon, J. Jo, A. Lodi, and Y. Bengio. Parameterizing branch-and-bound search trees to learn branching policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3931–3939, 2021.

J. Zou, S. Ahmed, and X. A. Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175:461–502, 2019.