الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد – تلمسان –

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE

THESE

Présentée pour l'obtention du **grade** de **DOCTORAT 3ème Cycle**

En : Génie Industriel

Spécialité : Ingénierie Logistique

Par : MECHAACHA Abdelkader

Sujet

Intégration des décisions de conception et planification des systèmes de production reconfigurables
Integrating Design and Planning Decisions for Reconfigurable Manufacturing Systems

Soutenue publiquement, le **20 / 02 / 2025** , devant le jury composé de :

M MEGNOUNIF Abdellatif	Professeur	Université de Tlemcen	Président
M BELKAID Fayçal	Professeur	Université de Tlemcen	Directeur de thèse
M BRAHIMI Nadjib	Professeur	Rennes School of Business, France	Co- Directeur de thèse
Mme KOULOUGHLI Sihem	Professeur	Université de Tlemcen	Examinateur 1
M KAHLOUL Laid	Professeur	Université de Biskra	Examinateur 2
M BENNEKROUF Mohammed	MCA	ESSAT Tlemcen	Examinateur 3
M ARBAOUI Taha	MC	INSA Lyon, France	Invité

Integrating Design and Planning Decisions for Reconfigurable Manufacturing Systems, ©
February 2025

Author:
Abdelkader Mechaacha

Supervisors:
Prof. Fayçal Belkaid
Prof. Nadjib Brahimi

Institutes:
Abou Bekr Belkaid University, Tlemcen, Algeria
Rennes School of Business, Rennes, France

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

This work would not have been possible without all the help and support provided by others. During my time as a doctoral student in Abou Bekr Belkaid Tlemcen university in MELT laboratory in Algeria and Rennes School of Business (RSB) in France, I had the honor of meeting a lot of interesting people, many of whom became friends. To all of you, I hereby say: *Thank You for everything! This would not have been possible without you!*, However, there are some individuals who deserve special mention.

Firstly, I would like to thank my supervisor **Prof. BELKAID Fayçal** for all the help he provided in and outside the university. He has been a teacher, a mentor, and a friend to me, teaching me how to work, how to conduct research, and to always defend my ideas.

I also want to thank my co-supervisor from RSB **Prof. BRAHIMI Nadjib** for helping us achieve this collaboration. He have been of huge help for me, and who taught me to be pragmatic in research and in life, and never say No to an opportunity!. Also thank you for having me in the occasions I had the chance to visit RSB in Rennes and IMT in Nantes.

I also wish to express my sincere gratitude to **Prof. MEGNOUNIF Abdellatif** president of the jury, for the honour of accepting to preside over this jury, and for their esteemed role in evaluating my work.
I extend my heartfelt thanks to **Prof. KOULOUGHLI Sihem** for their careful review of my work, their insightful comments, and their valuable contributions to improving the quality of this thesis.
My thanks also go to **Prof. KAHLOUL Laid** from the University of Biskra for accepting to evaluate our work and travelling from Biskra to participate in my defence, and for their careful evaluation, insightful suggestions, and valuable feedback on this work.
I am also grateful to **Dr. BENNEKROUF Mohammed** for their thoughtful observations, detailed feedback, and the time dedicated to reviewing my work.
I would also like to express my gratitude to **Dr. ARBAOUI Taha** for accepting the invitation to join the jury, for their thoughtful observations, and for the time and effort dedicated to reviewing my work.

Furthermore, I also want to thank all the members of the *Manufacturing Engineering Laboratory of Tlemcen-MELT* PhD students, and laboratory members for the many discussions and valuable inputs along the way. I am particularly grateful to **Radhwane** for helping us enhance the quality of the work of chapter 5. I would like to also thank all my teachers from the industrial engineering department. A special thanks also goes to the RSB team for their support of my application, especially **Prof. Sarah Hudson** without which this collaboration would not have been possible.

I also want to thank my parents, sisters and brother, special thanks to my mother who have supported me since day one, without her probably I wouldn't have made it through first grade. And of course last but not least, I want to thank all my friends and support team. A particular mention goes to **Hicham Abdelhak** my 3D designer, who drew the 3D parts presented in this thesis. And I would like to thank my two friends **Younes** and **Mohamed El-Hadi** for helping me during my trips to France.

PUBLICATIONS

This thesis includes work by the author that has been published or accepted for publication. These publications are the own work of the author of this thesis, and the author has the permission of the publishers to reproduce the contents of these publications for academic purposes.

In particular, some data, ideas, opinions and figures presented in this thesis have previously appeared or may appear shortly after the submission of this thesis as follows:

- Mechaacha, A., Belkaid, F., & Brahimi, N. (2024). "**Optimisation of process plan generation for reconfigurable manufacturing systems: efficient heuristics and lower bounds**". International Journal of Production Research, 1-23. ISSN: 0020-7543, DOI:10.1080/00207543.2024.2360089

- Mechaacha, A., & Belkaid, F. (2022, May). "**A meta-heuristic based approach for solving the single-item lot-sizing problem with a flow shop with energy and environmental constraints**". In 2022 14th International Colloquium of Logistics and Supply Chain Management (LOGISTIQUA) (pp. 1-6). IEEE. DOI:10.1109/LOGISTIQUA55056.2022.9938117
- Mechaacha, A., & Brahimi, N., & Belkaid, F. (2023, February). "**Mathematical modeling of single and multi-unit process planning problem in a reconfigurable environment**". In 2023 24ème congrès annuel de la Association Française de Recherche Opérationnelle et d'Aide à la Décisio (ROADEF) (pp. 1-6). . DOI:roadef2023.sciencesconf.org/436778
- Mechaacha, A., & Brahimi, N., & Belkaid, F. (2024, April). "**Sensitivity analysis of multi-objective optimization metaheuristic in solving the Multi-Product process planning**". In 2024 25ème congrès annuel de la Association Française de Recherche Opérationnelle et d'Aide à la Décisio (ROADEF) (pp. 1-6). . DOI:roadef2024.sciencesconf.org/512871
- Mechaacha, A., & Belkaid, F., & Brahimi, N. (2024, November). "**An Improved Heuristic and NSGA-II with Q-learning for Integrated Process Planning and Scheduling in Reconfigurable Manufacturing Systems**". In 2024 International Conference of the African Federation of Operational Research Societies (AFROS) (pp. 1-6). IEEE.

Except where reference is made in the text of the thesis, this thesis contains no other material published elsewhere or extracted in whole or in part from a thesis accepted for the award of any other degree or diploma. No other person's work has been used without due acknowledgment in the main text of the thesis. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

ACRONYMS

01-LP	0-1 Linear Program
1MDV	One Main Decision Variable Model
2MDV	Two Main Decision Variables Model
AMOSA	Archived Multi-Objective Simulated Annealing
CAD	Computer-Aided Design
CAPP	Computer-Aided Process Planning
CNC	Computer Numerical Controlled
CRP	Capacity Requirements Planning
CV	Coefficient Of Variation
DML	Dedicated Manufacturing Line
DP	Dynamic Program
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
GHG	Greenhouse Gas
H1.0	Heuristic 1.0
H1.1	Heuristic 1.1
HV	Hypervolume
I-MOILP	Iterative Multi-Objective Integer Linear Programming
ILP	Integer Linear Programming
IPPS	Integrated Process Planning And Scheduling
LB	Lower Bound
LB1	First Lower Bound
LB2	Second Lower Bound
MILP	Mixed-Integer Linear Programming
MOEA/D	Multi-Objective Evolutionary Algorithm Based On Decomposition
MOQ	Minimum Order Quantity
MPPP	Multi-Product Process Planning
MRP	Material Requirements Planning

MUPP	Multi-Unit Process Planning
NBI	Normal Boundary Intersection
NBI-es	Normal Boundary Intersection With Enumerate And Solve Strategy
NS	Number Of Solutions
NSGA-II	Non-Dominated Sorting Genetic Algorithm-II
PP-RMS	Process Planning For Reconfigurable Manufacturing Systems
RM	Reconfigurable Machine
RMS	Reconfigurable Manufacturing System
RMT	Reconfigurable Machine Tool
SILSP	Single-item Lot-sizing Problem
SM	Spacing Metric
SNR	Signal-to-Noise Ratio
SUPP	Single-Unit Process Planning
TAD	Tool Approach Direction
TOPSIS	Technique For Order Of Preference By Similarity To Ideal Solution
TSP	Travelling Salesman Problem
UB	Upper Bound
W-W	Wagner-Whitin

GENERAL INTRODUCTION

In todays fast-changing market, characterized by a growing demand for personalized products, customers seek a wide variety of options. At the same time, market opportunities are becoming shorter as customers rapidly shift their preferences and switch products. This dynamic environment presents significant challenges for manufacturing companies, leading to unpredictable fluctuations in both the volume and diversity of products demanded. Thus, to remain competitive and sustainable, companies need manufacturing systems that can adapt rapidly. As a result, many have turned to Reconfigurable Manufacturing System (RMS) as a viable solution.

Reconfigurable production represents a new type of production system designed for quick and efficient capacity adjustments, conversions, and the introduction of new products. This is achieved through the use of modular production equipment and close coordination between the product and production platforms, which became possible thanks to the development in industry 4.0 technologies.

In traditional fixed manufacturing systems, the design of the production system is established first, followed by production planning in a hierarchical manner. In production planning the assumption is that the system design is separated from production planning. However, in RMS, design and production planning decisions are inherently interconnected. Therefore, the objective of this thesis is to develop methods for efficient production planning, enabling effective use of RMS's "*reconfigurability*" potential.

The research is structured around three main areas: the first centres on the process plan generation problem in an RMS, the third emphasizes production planning by solving a lot-sizing problem with RMS, and the second bridges the gap between these two, by integrating aspects of both in a reconfigurable environment.

The objective of the first project is to solve a process planning problem with Reconfigurable Machine Tool (RMT). Process planning is an activity that typically follows the product design phase and precedes the actual manufacturing process. During this phase, the production process is broken down into smaller, more manageable tasks (operations), and the best methods for carrying out each task are determined. The objective is to select the most appropriate RMTs depending on the part's operations and their requirements, and assign the operations to the RMTs and determine their processing sequence. The objective is to minimize the total production time.

For this problem, three solution approaches are proposed: Mathematical models, Heuristics, and Lower Bounds.

In our second project, we have studied the integration of process planning and product sequencing decisions, product sequencing is a scheduling decision which is not typically made in the process planning step. We studied the problem in a multi-objective context, with the objectives of minimizing the total production time, and total production cost. We have employed two metaheuristics, namely Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D). We also developed a mathematical model that we implemented in Normal Boundary Intersection (NBI) approach.

For our last project, we opted for a single-item single machine lot-sizing problem with RMS, modelling both the Capacitated and Uncapacitated cases with consideration of startup costs and Minimum batch size.

After developing mathematical models for the two variants of the problem, a Dynamic Program (DP) that runsin $O(T^2C))$ time, with T being the number of periods and C Number of configurations is proposed and validated. The remainder of this thesis is organized as follows.

Chapter 1 was designed to familiarise the reader with the concept of RMS and its importance in the competitive landscape of today's markets and industry 4.0 revolution. In light of the distinctions introduced by these emerging systems, we underscore the necessity for adapting production planning methodologies, with a particular emphasis on process planning as the primary activity concerned. This activity is defined and its importance established by demonstrating its relationship with other production planning activities and with RMS. Subsequently, the distinction between production and process planning in the literature is clarified, and some production planning activities are introduced, with particular emphasis on lot-sizing, which is a fundamental element of production planning. Finally, the significance of integration between process and production planning is emphasised in the context of RMS.

Chapter 2 is more oriented towards operations research, with a focus on different optimisation methods. It begins with mono-objective optimisation and goes on to discuss two types of methods: exact and approximate. It also presents a number of methods, with an emphasis on those used throughout the thesis. Subsequently, multi-objective optimisation is presented, with an overview of the most widely used methods and performance indicators for evaluating the quality of solutions.

Chapter 3 presents a selection of key works related to process planning, process planning integrated with production planning, and lot sizing. It then introduces the relevant literature on RMS before exploring both process and production planning within this field. The integration of production and process planning within the RMS framework is also highlighted.

Chapter 4 details the process planning problem with a single unit, solved within an RMS, and presents the developed methods, which consist of a mathematical model, heuristics, and lower bounds. The chapter concludes with a discussion of the numerical tests that were conducted and managerial insights are presented.

Chapter 5 outlines the process planning with multiple products problem, with the proposed multi-objective solution approaches and algorithms. The NBI is initially presented, accompanied by the techniques developed for updating their β values iteratively. Subsequently, the enhancement method, which enumerates and solves, is presented before delving into the evolutionary algorithms. These include MOEA/D and NSGA-II, along with their crossover and mutation operators. The chapter is concluded with extensive tests regarding parameter tuning, small-sized and larger-sized instance testing.

Chapter 6 details the Lot-sizing problem with consideration of reconfigurable machines as well the different theorems used for developing the DP and proposed mathematical models for solving the two cases of the problem. Finally, it presents the conducted tests for comparing the Integer Linear Programming (ILP) with the DP.

Chapter 7 concludes this thesis. First, we summarize our work. Then, we highlight interesting aspects that we did not fully cover and provide a thorough discussion of potential topics and areas, with respect to process and lot-sizing and the integration of process and production planning all within a reconfigurable context, that may be interesting for future research activities.

1 RECONFIGURABLE MANUFACTURING AND PRODUCTION PLANNING

This chapter introduces fundamental notions and begins by discussing the context in general, considering the impact of Industry 4.0 technologies and customer behaviour shifts towards personalisation. It then introduces the *reconfigurability* paradigm as a solution to these new market needs, before discussing the two activities: process planning and production planning, with an emphasis on the impact of RMS on these two functions.

1.1 INTRODUCTION

Let us begin with a broad definition of manufacturing. The term "*manufacturing*" is defined as the process of transforming raw materials into finished products through the utilisation of various processes, equipment, operations and human resources, according to a comprehensive plan which is both cost-effective and generates income as a result of sales. These sales are achieved through the addition of value to the products sold by a company. Accordingly, the value added to a product can be defined as the increase in its market value resulting from modifications to its form, location, or availability, with the exclusion of the costs associated with the materials and services utilized (Scallan 2003).
This leads us to the definition of a manufacturing system, which is the system used to add value to products. In other words, a system in which raw materials are processed from one form into another, thus gaining a higher or an added value in the process, and thereby creating wealth in the form of a profit to the company. Today, these diverse manufacturing systems are undergoing a transformation with the introduction of new technologies driven by Industry 4.0. So, what is this new wave of change, and how does it impact manufacturing systems?

1.1.1 *Industry 4.0*

Industry 4.0 represents the fourth industrial revolution, which marks a fundamental shift in production paradigms and, consequently, influences various aspects of human life (Klingenberg et al. 2022). Historically, three prior industrial revolutions have resulted in significant changes to manufacturing through the introduction of groundbreaking inventions, each of which has had a profound impact on the methods used by humans to produce goods.

As stated by Gilchrist, 2016, in order to comprehend the nature of Industry 4.0, it is essential to have an understanding of the underlying technical foundation. That's why in his book he focuses on the emerging technologies that serve as enablers for the full potential of Industry 4.0, including artificial intelligence, augmented reality, 3D printing, and big data among others. As will be discussed in further detail, *reconfigurability* is closely aligned with the core principles of Industry 4.0. However, an industrial revolution is not merely a technological phenomenon, it arises within a specific context, which can be defined as "*the influences and events related to a particular event or situation*" (Klingenberg et al. 2022). It is therefore essential to consider the context in which these technologies have evolved, as it did so in response to human needs, not in isolation.

In their book "*The Global Manufacturing Revolution*", Koren, 2010 explores the recent landscape of customer behaviour, the one in which Industry 4.0 should operate, which is effected by globalization. He describes how shifts in customer behaviour drive today's focus on "*personalized production*".

1.1.2 *Personalisation*

The relentless pursuit of innovation has resulted in the creation of a dynamic and competitive marketplace, offering consumers a diverse range of options and the ability to purchase products from anywhere in the world via online platforms. In this context, businesses face the challenge of enhancing customer value by developing personalised solutions (customised products) while maintaining cost-efficiency achieved through economies of scale. The increasing complexity of customer demand is further forcing companies to diversify their supply chains and expand their product portfolios in order to gain and sustain a competitive advantage (Andersen et al. 2016; Rothaermel et al. 2006).

Industry 4.0 provides solutions to these challenges, reducing time to market and enabling the production of personalised goods at a low price.

The motivation for personalised production is not only driven by the need to address customer requirements, increase market share or respond to competitive pressure, but one of the key benefits of customisation is that it has the potential to create a diverse and large customer base. Such customer base generates more stable cash flow than a large homogeneous customer base, due to the greater stability of cash flow from a large number of customers with diverse tastes (Koren 2010).

While an exclusive focus on expanding a product portfolio may serve short-term objectives, achieving long-term success necessitates a strategic evaluation of the interrelationships between business, product, and manufacturing, coupled with an understanding of the development and maturity of Industry 4.0 technologies that have the potential to revolutionise manufacturing processes and product architecture. Furthermore, an examination of past production paradigms can provide valuable insights that enhance this analysis (for a detailed historical perspective, see Koren, 2010).

In this context, new product design approaches have emerged, such as modular products—smartphones being a prime example—where interchangeable modules allow for various combinations. This design concept introduces challenges extending beyond product development and into the design of manufacturing systems themselves (Campos Sabioni et al., 2022).

1.1.3 *Traditional Manufacturing systems*

In a more rigorous definition than that provided in section 1.1, A manufacturing system can be defined as:

> *A collection of manufacturing machines (or stations) that are integrated to perform a controlled set of repeatable operations on raw materials, which alter that material to achieve a desired final form, or to assemble a final product (Figure 1.1).* Koren, 2010

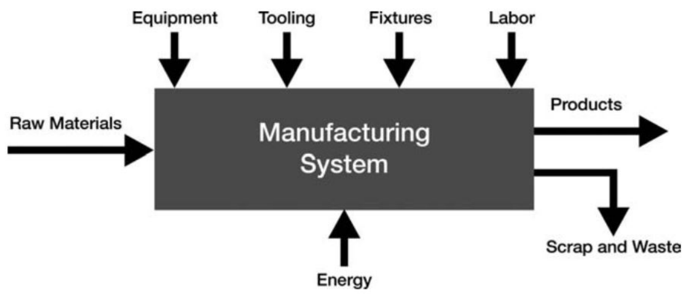


Figure 1.1: A manufacturing system converts raw material to a useful part or product (Koren 2010).

By the late twentieth century, the two most common types of manufacturing systems were the Dedicated Manufacturing Line (DML) and the Flexible Manufacturing System (FMS) [1].

FMS consists typically of Computer Numerical Controlled (CNC) machines that

1 Many industries used a mix of both dedicated and flexible manufacturing systems also.

are arranged in parallel. These systems are designed to produce a diverse range of products, though in relatively low volumes. In contrast, DMLs are composed of dedicated machines aligned in production lines, each line dedicated to a single product type. This configuration offers high throughput but limited product variety. As illustrated in Figure 1.2, FMS provides high variety and low-volume production, while DML offers low variety but high throughput.

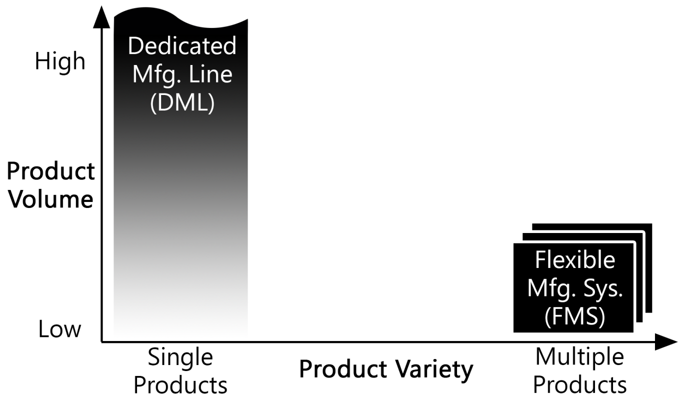


Figure 1.2: Comparison between FMS and DML (Koren 2010).

A report published in Italy in 1998 revealed that in a European automotive manufacturing company, the average utilisation of DMLs was only 53%. This low utilisation often results from a reduction in demand at the early or late stages of a product's lifecycle, when production volumes fall below the optimal levels forecasted during line design. Furthermore, due to global competition, even mature products often fail to reach the high production volumes initially projected.
While the main problem with the high-throughput DMLs, was that it lacked flexibility, FMS were a natural answer to address this limitation, especially as the world at the time shifted beyond the era of mass production toward mass customisation. FMSs (typically composed of parallel CNC machines, as mentioned earlier) deliver flexibility in both variety and, if feasible, capacity through the addition of CNC units. However, the cost and time required to maintain this flexibility—especially unused capacity—has driven industry toward a manufacturing system that can provide, in Koren's words, *Exactly the capacity and functionality needed, exactly when needed* Koren, 2010. RMS stand as a balanced solution, bridging the limitations of both DML and FMS to meet evolving production demands.

1.2 RECONFIGURABLE MANUFACTURING SYSTEMS

The novel manufacturing system, RMS, is designed to address the shortcomings of traditional manufacturing systems by improving reactivity to changing market conditions and facilitating rapid and cost-effective adaptation in an unpredictable market context (Andersen et al. 2023; Koren et al. 2018). It is defined by Koren, 2010 as:

> *A RMS is one designed for rapid adjustment of production capacity and functionality, across a product family, by rearrangement or change of its components (hardware and software).*

On closer analysis, it can be seen that the definition provided offers a precise description of what is meant by RMS. To gain a deeper understanding, it is necessary to break it down.
The first part with the term *"designed"* indicates that the concept of reconfigurability is an embedded feature of an RMS. It is not a consequence of strategic thinking about how to reconfigure a given system, rather, it is an inherent attribute that must be intentionally incorporated into the system's design.
The term *"rapid adjustment"* indicates that an RMS is designed in such a way that the changes it can support must be implemented rapidly. Otherwise, the adaptability of the system would be of no benefit to a manufacturing company if it were not implemented in a timely manner when required. In fact, if the necessity for rapid adjustment is not considered, any system can be modified to adapt to volatile markets. However, only RMS is capable of doing so in a timely manner.
"of production capacity and functionality" indicates that the adjustment can be made in two ways: first, in terms of capacity which means that the throughput[2] can be modified from one period to the next, second, in terms of functionality which means the type of product or operation being processed can be altered from one period to the next. The aforementioned attributes represent fundamental characteristics of RMS.
The term product family refers to a group of products that share similar features or operations. In this context, it signifies that the adaptability of the RMS system is not unlimited. It is not feasible to change from manufacturing any product to manufacturing any other product, instead, the change is made within a product family. This is another defining characteristic of RMS.
The last part of the definition explains how this change is achieved: it is done by rearranging or changing the components of the system. These components may include the placement of machines, the components of the machines themselves, material-handling equipment, and so on.

2 Throughput is the production rate for a given period of time (e.g., the hourly or daily production rate) (Swamidass 2000).

1.2.1 *RMS Characteristics*

The author who coined the term RMS in Koren et al., 1999, identified six key characteristics that an RMS must possess, namely scalability, convertibility, customization, modularity, integrability, and diagnosability, of which the first three are fundamental.

- Customisation: The ability to use customised flexibility in production to meet new requirements, however, within a certain product family.
- Scalability: The ability to modify or expand system or machine components to effectively change production throughput.
- Convertibility: The ability to effectively change the functions of the machine or system to meet evolving production needs.
- Modularity: The division of hardware and operational functions into components that are changeable in different machine or system configurations.
- Integrability: A set of mechanical, informational and control interfaces that allow these modules to be integrated quickly and accurately.
- Diagnosability: The ability to track the current state of a machine or system and its controls in order to identify and diagnose the root cause of defects in the end product.

In order for an RMS to be cost-effective, it must possess the first three essential characteristics, namely convertibility, scalability and customisation. In comparison to a system with generic flexibility, an RMS and the associated machines with customised flexibility will be less expensive to build and operate. Similarly, a comparable FMS would take more time to process the same operations. Meanwhile, the characteristics of modularity and integrability are sufficient to construct an RMS. Lastly, when incorporated into the machine and its control system, diagnosability provides the capability to rapidly and accurately reconfigure the system.
An important insight derived from the description of these characteristics is that reconfigurability must be considered and planned for from the outset, both in terms of the system and its machine design. Otherwise, the process will be lengthy and impractical (Haddou Benderbal et al. 2017).

1.2.2 *RMS vs FMS vs DML*

To better understand the usefulness of RMS, it is better to compare it directly to the traditional manufacturing systems. As mentioned in previous sections, an FMS is not as cost-effective as DML. However, it does provide the flexibility required in order to transition between manufacturing product variations. In contrast, an RMS combines both the productive capacity of DML with the flexibility of FMS. It offers flexible, cost-effective manufacturing and adaptable structure (both machine-level and system-level) to manage unexpected changes in the market.

As figure 1.3 indicates, DMLs offer a high productivity with a low variety (usually manufactures only one type), and in contrast, FMSs are a high variety, low productivity manufacturing systems. RMS, unlike the other two, can switch from one product type to another and from one capacity rate to the other through its life cycle. Not just that, but Koren, 2010 claims that reconfiguration allows an RMS to even achieve throughput approaching that of a DML and producing simultaneously several products.

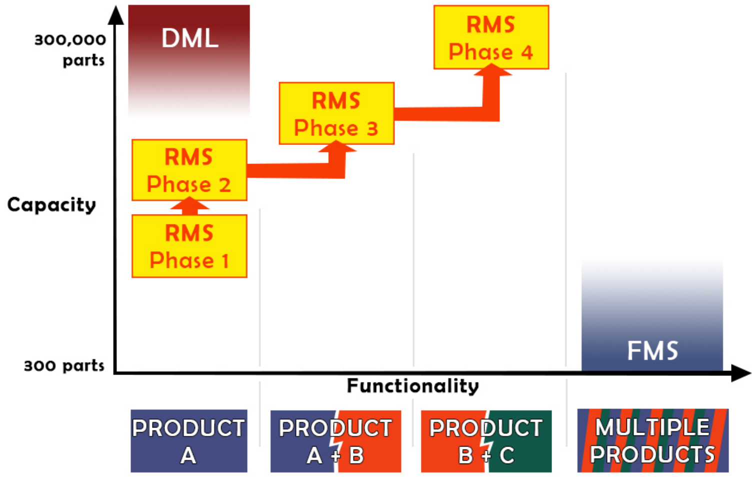


Figure 1.3: Comparing the three manufacturing systems Koren, 2010

The Reconfigurable Machine (RM) and its most popular example, the RMT, represent the fundamental components of the RMS (Gadalla and Xue 2017). The principal differentiations between an RMT, a CNC machine, and a dedicated machine of DML are presented in Table 1.1.

	Dedicated	RMT	CNC Machine
Machine structure	Fixed	**Adjustable**	Fixed
Design focus	Part	**Partfamily**	Machine
Scalability	No	**Yes**	Yes
Flexibility	No	**Customized**	General
Simultaneously operating tools	Yes	**Yes**	No

Table 1.1: Comparing the three types of machines Koren, 2010

Just like an RMS, an RMT is built differently from ordinary machines. An RMT is conceived to be reconfigurable thanks to its adjustable structure. Its design evolves around a part family, and it permits change in capacity or functionality (scalability and convertibility), However, this flexibility is limited to the part family members and therefore, only provides the flexibility needed for those specific parts (customization).

The edge that dedicated machines have over CNC machines in terms of productivity is mainly due to the focus on the part at the design stage, which enables opportunities for enhancing system productivity. However, thanks to the part family being the design focus, RMTs can also take advantage of these opportunities, such as simultaneous multiple-tool operations, e.g., gang.

1.2.3 *Flexibility vs Reconfigurability*

While the distinction between DML and RMS is clear, the same cannot be said for the distinction between FMS and RMS. Accordingly, this subsection will focus on the difference between flexibility and reconfigurability.

In response to this question, Wiendahl et al., 2007 offers a simple yet compelling answer. In accordance with their definition, flexibility is typically understood to be the capacity of a system to change its behaviour without modifying its configuration. While reconfigurability is the capacity of a system to change its behaviour through modification of its configuration. It can be seen that the application of these concepts is only possible when the system's boundary is clearly defined. Thus it is possible to understand a specific form of changeability as either flexibility or reconfiguration, depending on the boundary in question.

Furthermore, Andersen et al., 2023 provides an additional clarification through the illustration presented in figure 1.4.

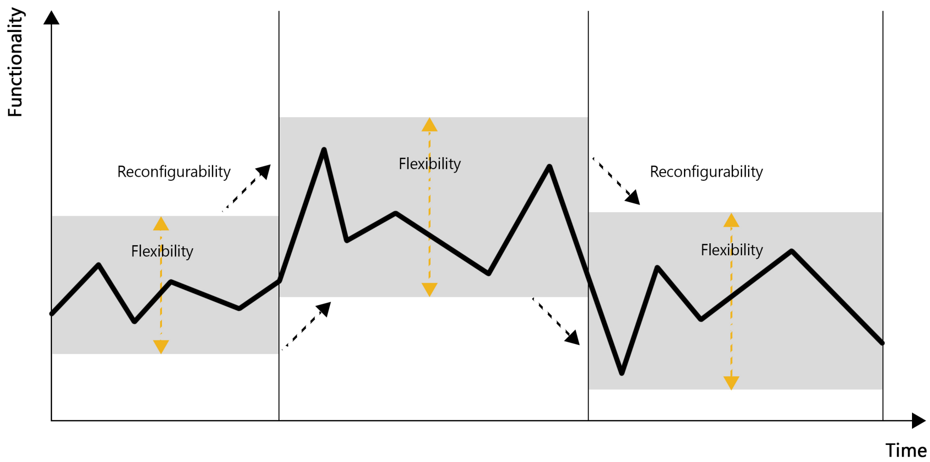


Figure 1.4: The difference between flexibility and reconfigurability (Andersen et al. 2023).

The primary distinction between FMS and RMS is in the design phase, wherein the former prioritizes expanding the system's production range, whereas the latter emphasizes adaptability and responsiveness to changing demands. The reconfigurable manufacturing system prioritises customised flexibility over general flexibility, with the objective of enhancing the system's capacity to adapt to a variety of requirements and circumstances (Qin et al., 2016).

These changes in capacity or variety are made either *i*) by reconfiguring the system or *ii*) by reconfiguring its machines.
At system level, the RMS architecture permits the addition or removal of machines in a cost-effective and timely manner. X. Gu and Koren, 2018 proposed an example of such architecture (Figure 1.5) to cost-effectively produce mass-individualized products. The products can move in both directions, allowing return to previous stages using, for example, Autonomous Guide Vehicles (shown as a red line in the figure), and machines can be added and removed as modules to stages.

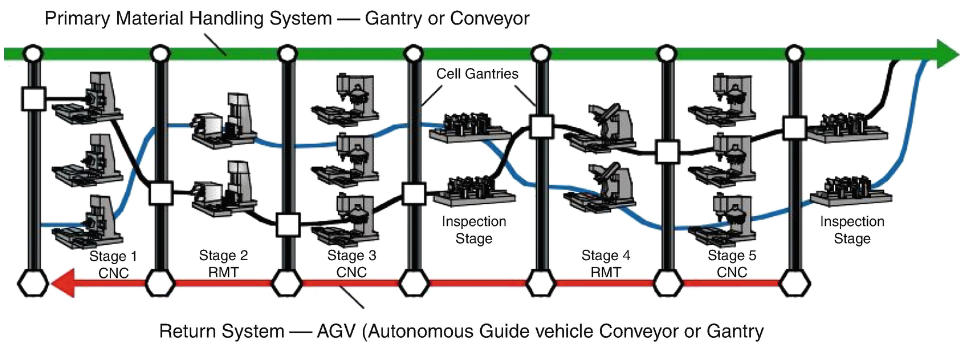


Figure 1.5: An example of the architecture of an RMS X. Gu and Koren, 2018.

1.2.4 *Reconfigurable Machine Tool*

At the machine level, RM is the component that enables the reconfigurability potential of RMS. To adapt to evolving operational requirements, an RM may undergo modifications to its structural configuration in order to provide either incremental increases in production throughput or alternative functional capabilities. Then when required, the RM may be modified to incorporate additional features or enhance manufacturing capacity, or it can be returned to its initial state.
An RM may be built for various manufacturing applications:

- Machining: reconfigurable machine tool (RMT): As its name indicates, an RMT is a machine tool that can be reconfigured. A machine tool is a machine that performs deformation operations on various materials using a tool (Aguilar et al. 2013).

- Assembly: reconfigurable assembly machine: These machines are used in assembly lines and can be adapted for products belonging to the same family. An example of an assembly machines for automotive heat exchangers is provided in Katz 2007.

- Fixtures: reconfigurable fixtures: Used for fixing parts during machining, typically for complex parts like engine cylinder heads. Complex fixtures cost around $30,000, and in industrial machining, there are 50 to 100 fixtures for each part, and they must be replaced once the product is changed (Koren 2010). Therefore in high-mix machining, this investment is considerable.

- Inspection: reconfigurable inspection machine: These machines are used to inspect and measure parts' quality. Their number and location can be changed to suit the part being examined (Shang et al. 2021).

In our work the main focus is on RMTs. An RMT is reconfigured by changing its hardware or software components. By changing its configuration, an RMT can be used as a group of machines. Different configurations can be created by assembling and disassembling its modules during the operation stage.
Aguilar et al., 2013 provided a design of an RMT that can be used in jewellery industry. They stated that the conversion between configurations can be completed in less than 15 min. They also tested the prototype and provided a video of the RMT (The video is provided as supplementary data and is available on the publisher's website). A review on RMT-related literature including its architecture design can be found in Gadalla and Xue, 2017.
The difference between an RMT and an ordinary machine tool is that an RMT has several configurations, each with a different functionality or capability. The assumption that RMTs are by definition modular is a relatively common but inaccurate notion in the literature. While modularity is a design element, it is not an obligatory component of RMT. Koren himself presents a non-modular RMT, the so-called 'Arch type' in Koren, 2010; Koren et al., 1999, which is illustrated in Figure 1.6. The machine's configuration can be changed by adjusting the angle of the spindle.

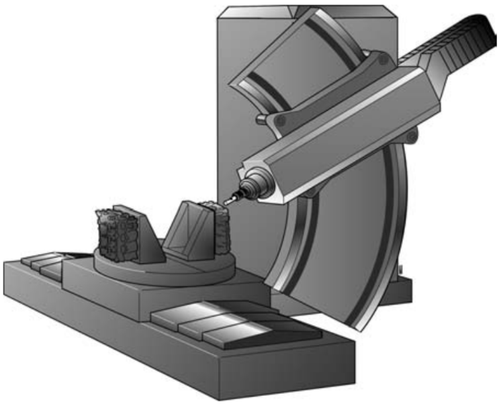


Figure 1.6: An Arch-type RMT (Koren 2010).

Figure 1.7 illustrates an example of a modular RMT, which is composed of modules that serve specific purposes. Some modules are basic modules, indispensable for the RMT to function in any configuration. Others are auxiliary modules, necessary only for certain configurations (Haddou Benderbal et al. 2018).

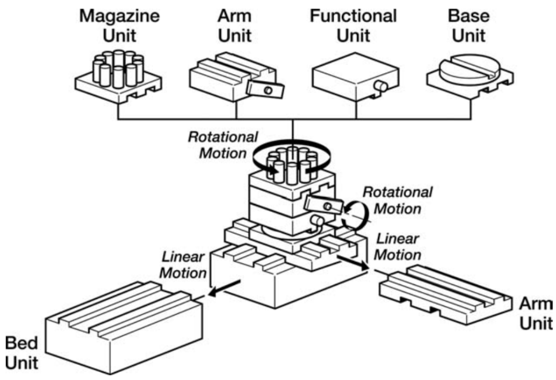


Figure 1.7: A modular Reconfigurable Machine Tool (Koren 2010).

As indicated in Table 1.1, the cost-effectiveness of RMTs is achieved through their design, which is tailored to a specific range of operational requirements, and their ability to be economically converted from one operation to another. The challenge is to concentrate the machine design effort on a specific part family to create an adjustable machine capable of machining features of every part in the family in a timely manner. The degrees of freedom of an RMT are designed based on the operational requirements of all parts in the family. As operational requirements change, the RMT must undergo mechanical modifications to adapt to these changes.

With this difference brought by RMTs, production planning must cope with this new "reconfigurability" paradigm, and as its design depends heavily on the part family operations and their requirements, the generation of process plans function of these parts becomes also important. In addition, looking at the potential of

RMTs, these machines must be used in the most cost-effective manner. This means that the manufacturing of the products must be planned to make the best use of these resources and their "reconfigurability" potential, which is the very essence of process planning (Scallan 2003). These two points are discussed further in sections 1.3 and 1.4.

1.2.5 *Industry 4.0 and RMS*

In relation to Industry 4.0, Qin et al., 2016 conducted a review of the key characteristics of existing manufacturing systems (including FMS and RMS) with a view to identifying areas of research that are currently lacking in relation to these systems and industry 4.0. The authors concluded that RMS is the most promising production system for achieving the objectives of Industry 4.0, including the capacity for self-optimisation and self-configuration. Furthermore, a key objective of Industry 4.0 is the production of individualised products at a reasonable cost, a paradigm known as personalized production (See section 1.1.2), as previously stated, this is a core objective of RMS.
Figure 1.8 is derived from Qin et al., 2016. The author stated that existing manufacturing systems are already capable of addressing several aspects of Industry 4.0, particularly those related to digitalisation and communication. Nevertheless, it is challenging to claim that these systems have reached the level of intelligent manufacturing envisioned by industry 4.0, especially that they have not yet attained its lower or upper levels of cognition and configuration, respectively. These gaps represent the primary focus of Industry 4.0 future research.

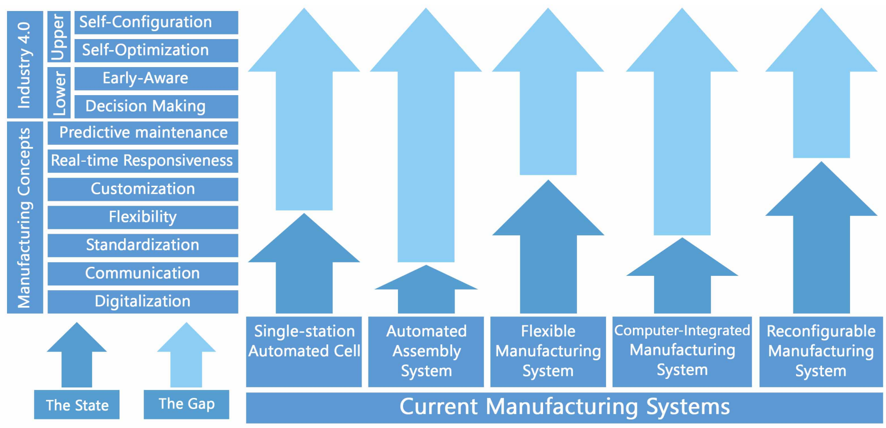


Figure 1.8: Research gap between current manufacturing systems and Industry 4.0 according to Qin et al., 2016.

Because Industry 4.0 is primarily concerned with automation, smart factories, and interconnected systems driven by cyber-physical systems and the Internet of Things, its primary objective is to enhance process efficiency. Consequently, it

overlooks the human cost associated with the optimisation of processes. This is the primary issue that will become apparent in the near future, as the full impact of Industry 4.0 becomes evident (Nahavandi 2019). In fact, one might argue that such effects are already in place, given the recent changes in weather and instability.
Accordingly, researchers anticipate that the forthcoming Revolution or *Industry 5.0* will provide a solution to the issues currently being or that will be experienced by Industry 4.0. This new paradigm will extend the scope of previous models to encompass *human-centricity*, *resilience* and *sustainability* within industrial operations. This focus on human involvement, particularly through collaborative robots and customised production, is well aligned with RMS, which is designed to facilitate frequent reconfigurations in response to changing product demands and production conditions. And given that RMS enhances resilience within enterprises, it makes it relevant in an Industry 5.0 context. For further details on this new revolution, please refer to Nahavandi, 2019 and Xu et al., 2021.

1.2.6 *Real world applications of RMS*

Many practitioners and even academics find it challenging to envision RMS in real-world applications. "A system capable of adjusting its capacity and functionality both cost-effectively and rapidly" sounds ideal, which leads to skepticism. However, examples of RMS principles implemented in manufacturing exist and illustrate its potential.

One example comes from the SmartFactoryKLs soap plant (Qin et al. 2016). Built in 2005, the plant became a pioneer of Industry 4.0, producing customizable liquid soap in various colors without human intervention. Machines and components communicate autonomously, and the modular structure allows components to perform defined functions, eventhough the design focus is on industry 4.0, RMS principles are clearly embedded.

The book Andersen et al., 2023 "*Paving the way for changeable and reconfigurable production*" describes the REKON project, developed to help Danish and Swedish companies integrate reconfigurability and changeability. The book provides real world applications in companies like Vestas Wind Systems and Dan-Foam ApS which used these principles to adapt their production lines for customization and demand fluctuations.

- Vestas designed modular equipment/machines for wind turbines production which has to be tailored to specific environments, this increases variance in their product portfolio. Their main focus was on optimizing the reconfiguration process.

- Dan-Foam developed a reconfigurable production line that minimized changeover time and adapted to demand changes. In addition, the company was able to replace two existing production lines with the new one.
- Ljusgårda AB, an innovative indoor farm in Sweden, applied principles developed by REKON for scalable and flexible lettuce production. The company identified bottlenecks using simulations, improving their production systems efficiency.
- Similarly, Kamstrup A/S, an energy and water metering solutions manufacturer, used REKON tools to map equipment across its different factories, identifying reusable processes and equipment to support modular production.
- Elvstrøm Sails, a sail manufacturer, developed a tool with Aalborg University to optimize assembly, reducing lead times by 45%.
- Lastly, Hydrema A/S, a construction machinery producer, created modular welding fixtures, enhancing equipment reusability and efficiency. This fixture reduced the need for dedicated tools and streamlined production.

These examples showcase RMSs real-world applicability, proving that changeability and reconfigurability are achievable and beneficial for various industries. For more real world examples and more details about the mentioned ones please refer to Andersen et al., 2023.

Although the concept of an RMS is clear, the challenge remains as to how it can be effectively utilised or even designed in order to meet specific production requirements. This brings us to the crucial role of process planning, which defines how RMS configurations can be structured and adapted for optimal performance.

1.3 PROCESS PLANNING

The term "*process planning*" is defined in its simplest form by P. Gu and Norrie, 1995 as "*the act of preparing detailed operating instructions for the conversion of an engineering design into an end product*." It is also defined by Scallan, 2003 as the "*Design-Manufacture interface*," as presented in Figure 1.9, it represents the bridge that links the design of the product phase with the actual manufacturing of the product.

During this phase, the production process is broken down into smaller, more manageable tasks (operations), and the best methods for carrying out each task are determined. A process planner often attempts to answer two questions: How to manufacture the given product? and How long it would take to execute the operations and the manufacturing process?

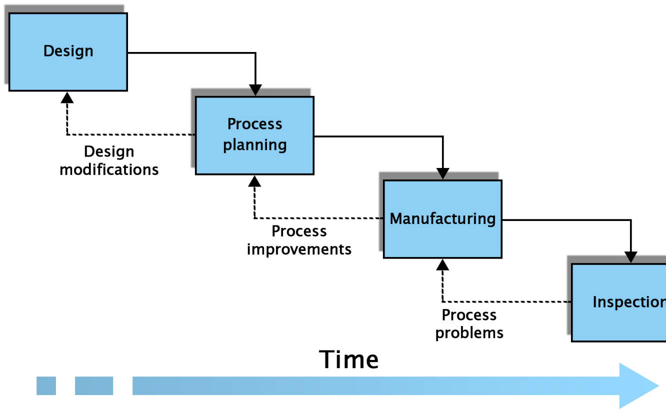


Figure 1.9: Process planning - the design manufacture interface (Scallan 2003)

Moreover, as Larsen, 1993 notes, process planning is a more technical activity, encompassing tasks that are mostly time-independent. It is linked in the long-term to overall factory planning. At the medium level, individual goods are planned in a process-oriented approach, with an analysis and selection of materials, production techniques, and resources. In contrast, the detailed level of process planning addresses tooling, fixturing, and the sequence of operations. The final tasks include the calculation of resource requirements (processing time, setup, etc.) and the creation of NC programs (if necessary) and production documentation.
Larsen, 1993 lists the process planning related tasks from the highest level to the lowest, in the following order:

- Material selection.
- Process selection.
- Machine selection.
- Operations selection.
- Tool selection.
- Process sequencing.
- Operations sequencing.
- Fixture selection.
- Machining parameters.
- Cutting path optimization.
- Cost and time estimation.
- Process plan preparation.

- Date maintenance.
- Plan verification.

Process planning is typically associated with machining processes (metal removal operations). This process planning domain of application is the most sophisticated, and in this thesis, focus is on this type. However, it should be noted that other process planning applications exist, including Assembly process planning, Inspection process planning, Robots Task Planning, Process Planning for Welding Operations (ElMaraghy 1993).

1.3.1 *Classifications of Process planning*

Prior to the integration of computers into industry, process planning was conducted manually by human experts. Such planners would rely on their expertise and human intuition to define the process plan for parts. In this regard, two approaches exist. The first is the traditional approach, in which the planner uses the part drawings and manuals to determine the processes to be employed and the operations to be executed. These are then documented in a routing sheet.
The second approach is that of the workbook. This approach involves the preparation of workbooks containing pre-defined sequences of operations for specified workpieces. Once the manufacturing processes for a specific part have been identified, the appropriate sequences can be selected and incorporated into the plan from the workbook.
One of the primary disadvantages of manual approaches is the generation of voluminous and inefficient documentation. Furthermore, the effectiveness of the plan is dependent on the level of experience of the planner. It is possible for two planners to utilise the same methods and yet generate different results. Furthermore, design modifications are frequently required towards the end of the design and manufacturing cycle. And, the manual approach lacks the capacity to respond to such changes in a timely manner (Scallan 2003).

In response to these disadvantages, a significant body of research has been conducted in the domain of Computer-Aided Process Planning (CAPP). The benefits of such CAPP systems are believed to include a reduction in the time spent on process planning, a decrease in the reliance on the knowledge and experience of the process planner, an enhanced utilisation of manufacturing resources, and most importantly improvements in costs.
CAPP can be further categorized into variant and generative. Variant approach just like the workbook approach relies on existing knowledge. It consists of retrieving an existing plan named "*Master plan*" which is usually a process plan of a representative part of the part family and numerically modifying it. The variant

approach is also named the "*retrieval method*". Despite being used widely in industry, one problem with it is that, plans can only be developed for parts already classified.
The second CAPP approach is the generative one, which consists of generating a new process plan from scratch. In theory, it requires no need for any form of human intervention. In practice however, this is often not the case, as for the majority of generative systems, some form of human intervention is typically necessary (Scallan 2003).
Figure 1.10 summarizes the described four categorizes of process planning.

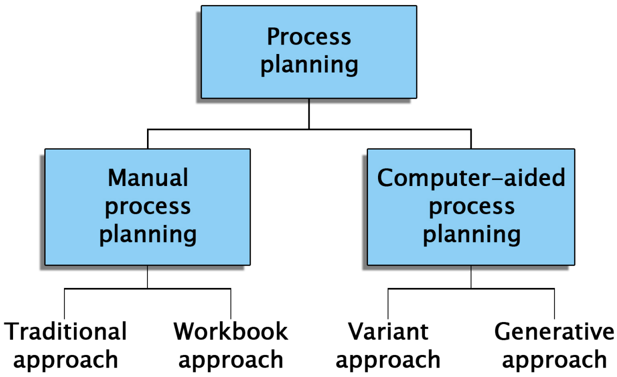


Figure 1.10: Basic classification of process planning methods (Scallan, 2003)

Process planning can be further categorized, depending on the level of the decision being made, in two distinct levels (Azab and ElMaraghy 2007b):

- Macro-process planning involves selecting the best sequence of processes, set-ups, and machines to accomplish the required operations.
- Micro-process planning, in which each operation is optimised individually to determine its optimal parameters, which is more detailed then macro.

Macro process planning represents a harder challenge due to the necessity of declarative process knowledge, which encompasses various factors such as part geometry, tools, machine tools, fixtures, and technological requirements (Azab and ElMaraghy 2007b).

The nature of process planning and the resulting plans are influenced by a number of factors. The objective of high-level planning is to select the optimal technique for producing a feature, component, or part. This may entail the use of processes such as metal removal, material addition, shaping, or joining, among others. The result is a generic or conceptual plan. These tasks of the selection of materials and processes represent the highest levels, as previously discussed in the aforementioned list of Larsen, 1993.
Additionally, the focus of the planning process may be on a single domain, such as

sheet metal processing or assembly, or it may encompass multiple domains. The latter is referred to as multi-domain process planning.
Alternatively, the planning process may focus on certain tasks within the broader process planning framework. The terms "*process planning*" and "*process plan generation*" are typically used interchangeably in the literature, although the former term is more commonly used to refer to macro-process planning, focusing mostly on operations sequencing with other tasks such as operations selection and machine selection. These two terms will be used interchangeably in the remainder of this thesis.

1.3.2 *Pre-processing and Tool approach direction*

Before starting low-level tasks of process planning such as operations selection and sequencing, several preparatory steps are required. Such as the manufacturing process to be used. Another key step is feature[3] recognition, where the Computer-Aided Design (CAD) of the part is translated into a set of operations grouped by features, with precedence constraints and tooling and Tool Approach Direction (TAD) requirements. This is illustrated in Figure 1.11, where three features are identified from an example part (Part 1) CAD and their operations are connected by a precedence graph with corresponding tools and TADs.

After these initial tasks, the feasibility and economic merits of alternative methods are typically overlooked during process plan generation. These steps, while important for context, are beyond the scope of this thesis and will not be further discussed.

The direction in which the tool moves towards the workpiece during a machining operation is referred to as the tool approach direction (TAD). Consequently, as a given feature is being machined, the tool is oriented in a specific direction and moves towards the workpiece (Rong and Huang 2005). By employing the conventional Cartesian coordinates ($\pm x, \pm y, \pm z$), the TADs for each operation can be determined, resulting in six distinct TADs as illustrated in Figure 1.12.

3 A machining feature is defined as a geometrical feature that requires processing by one or more operations (Azab and ElMaraghy, 2007b).

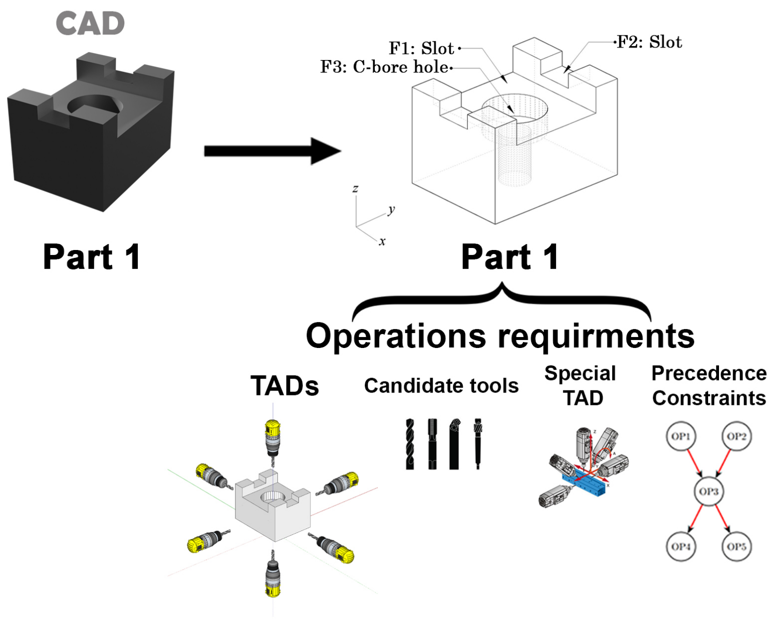


Figure 1.11: Pre-processing step required before process planning where operations requirments of a part are retrieved based on its CAD. (Mechaacha et al. 2024).

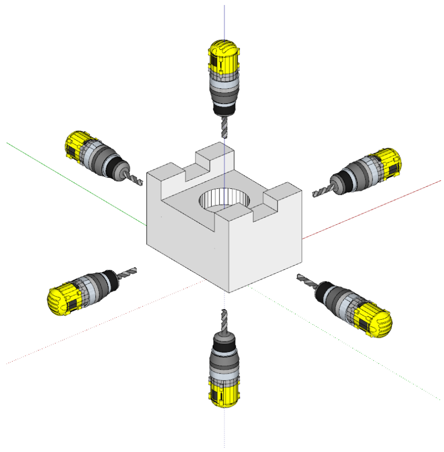


Figure 1.12: TAD (Mechaacha et al. 2024).

The initial example component (depicted in Figure 1.11) is derived from the work of S. Zhang and Wong, 2016 and will be employed in subsequent numerical experiments to evaluate the efficacy of the models (See chapter 4). Each feature is comprised of a series of procedures. Every procedure necessitates a minimum of one TAD, with some requiring multiple TADs.

The operations designated as F3 in Part 1 of Figure 1.13 require machining in a vertical plane, both in the positive and negative Z-axis TADs. Conversely, a variety of feasible TADs are available for the machining of slots F1 and F2. Part 2 of S. Zhang and Wong, 2016 is also depicted in Figure 1.14.

It is important to note that in the numerical experiments and models of chapter 4, the data has been modified based on the assumption that a machine requires all of the required TADs in order to be capable of performing an operation. This assumption, which is not widely employed in the literature, is due to the fact that our generated process plans do not specify one TAD for the machining of each operation. Instead, operations are assigned to machines and configurations, and the choice of TAD is left to the operator. This approach is suitable for scenarios where operators are sufficiently skilled and the manufacturing process is complex or involves a high degree of uncertainty.

In total, eight TADs are considered, as certain special operations require the use of particular TADs (Not the six conventional ones $\pm x, \pm y$ or $\pm z$ from Figure 1.12), which are defined as $\pm a$ (Azab and ElMaraghy 2007b; W. Li and McMahon 2007; S. Zhang and Wong 2016; Y. Zhang et al. 2003).

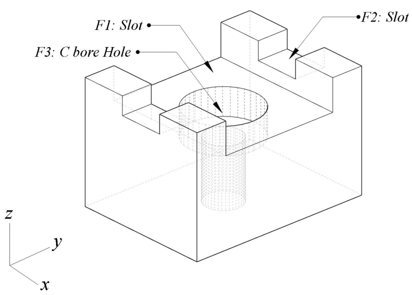


Figure 1.13: Part 1 (S. Zhang and Wong 2016).

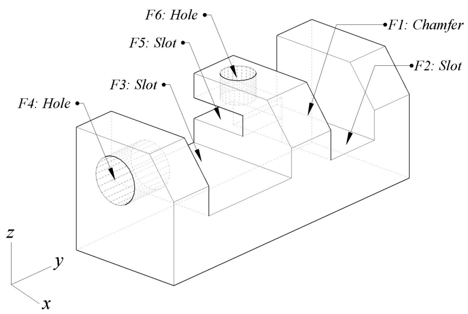


Figure 1.14: Part 2 (S. Zhang and Wong 2016).

1.3.3 *Importance of process planning and system design*

Process planning represents an essential component of any manufacturing enterprise, especially with the recent advances and interest in integrated manufacturing, product/process design, and life-cycle considerations (ElMaraghy 1993). Additionally, the preparation of the process plan[4], which constitutes the outcome of process planning, serves as the foundation for the subsequent planning and control of the manufacturing process for any given product (Larsen 1993; Scallan 2003).

As previously indicated, process planning tasks are of different levels (See the list in section 1.3). Consequently, process planning can be of strategic, tactical, and operational importance, depending on the type of environment employed (Rembold et al., 1993).

In a relatively stable market with a limited product portfolio, where DMLs would be best suited, the process planning is conducted only once, at the system design stage. Nevertheless, in markets characterised by greater volatility, process plan-

4 The French term for a process plan is "gamme de fabrication" or "gamme de production".

ning is conducted more frequently. (Bensmaine et al., 2013b).
In both scenarios, process planning, given its position between the design and manufacturing phases, plays a pivotal role influencing the decisions related to the production system. For example, the process plan for a proposed product designed by the design team may present difficulties in terms of its manufacturability on the existing system. This would require an investment in machinery, layout adjustment, and material handling equipment. Similarly, a process plan may necessitate the modification of components within a machine. (Battaa et al. 2017; Toussaint and Cheng 2002).

In their study on CAPP for flexible assembly, Heemskerk et al., 1990 predicted that the distinction between product design, system design and process planning would gradually disappear. They also suggested that the development of faster process planning systems would facilitate rapid feedback to designers on the impact of decisions made during the product or system design stage on assembly cycle time and other optimisation factors.

It can be seen that process planning serves as a crucial enabling factor for changeable systems such as RMS from an operational perspective, as the process plan will affect both the product and the manufacturing system in terms of cost, quality and production rate (Scallan 2003). It is therefore crucial to integrate the configuration capabilities of RMTs into the process plan model in order to accurately represent a process plan for RMS (Azab and ElMaraghy 2007b; Shabaka and ElMaraghy 2008).

1.3.4 *RMS and process planning*

As previously discussed, RMS was developed to address the challenges posed by volatile markets. Given the unpredictable nature of production demands and fierce competition, it becomes important to respond rapidly to changes in the market. In such circumstances, any shortcomings in process planning are likely to have significant consequences (Musharavati et al., 2008). Moreover, the process plan in RMS must be revised frequently, given the constant evolution of equipment and products. The configurations of machines and the associated reconfigurations, along with the technical aspects taken into consideration, are determined by these process plans (Sabioni et al., 2021).
Typically, process planning with RMS entails identifying optimal machine configurations for operations, determining the best sequence for these operations and later selecting the best machine placement into the layout (Sabioni et al., 2021).

An additional issue in process planning is related to part designs being infeasible where no system adjustment can be made, while maintaining cost-effectiveness and timeliness. This is often due to designers lacking a comprehensive understanding of production system capabilities and constraints. This issue is further

amplified within complex manufacturing systems such as RMS.
While manufacturing teams possess the required knowledge in these matters, their ability to offer real-time or timely feedback to design teams after converting designs into process plans, encompassing total production time, cost, and quality considerations, may be limited. This is particularly true in collaborative environments where teams are geographically dispersed (Toussaint and Cheng 2002).
Consequently, the significance of generating fast solutions in process planning becomes evident, as this would permit for the designer to recognise, at an early stage, that certain features will necessitate the use of costly fixturing or the deployment of specialised tools and complex processes. In such instances, he can explore alternative solutions.

The process plan, while serving as a flexible input for various planning activities such as scheduling, lot sizing, and layout planning, often requires adjustments in these subsequent stages. This highlights the importance for consideration of production planning activities to ensure an effective design of process plans for RMS (Scallan 2003; Shin et al. 2011).

1.4 PRODUCTION PLANNING

The process plan generation phase represents the initial stage in the planning of any product manufacturing, as illustrated in Figure 1.15. Although the titles of section 1.3 and this section may suggest that process planning and production planning are distinct functions, process planning is, in fact, part of the production planning activities (Scallan 2003). However, as the term "production planning" is typically used in academic literature to describe time and capacity-related planning tasks (ElMaraghy 1993; Larsen 1993), some authors prefer to distinguish between the two terms.

As Figure 1.15 illustrates, process planning is a long-term decision. As previously stated in Section 1.3, the process planner's primary objective is to answer two questions: how a job should be done and how long it will take. Consequently, the availability of resources on the shop floor is not within the process planner's concern. For example, the process planner is not responsible for ensuring that workstations are not overloaded in terms of capacity. This is the responsibility of the Capacity Requirements Planning (CRP) function, which is also illustrated in Figure 1.15 (Scallan 2003).
Medium-term planning decisions presented in figure 1.15 are CRP and Material Requirements Planning (MRP). In practice, CRP, MRP and production execution are separated in terms of personnel, software, and time. The MRP system is typically responsible for establishing the necessary materials and generating a general schedule without consideration of capacity constraints. CRP determines the avail-

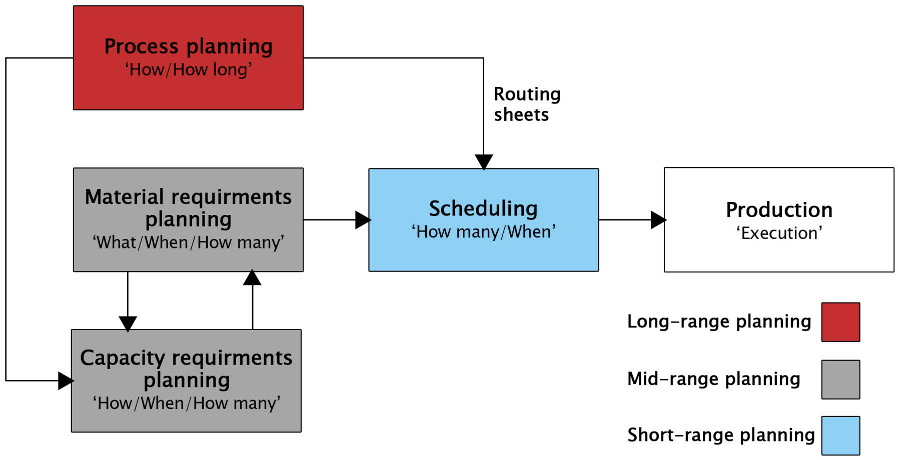


Figure 1.15: Production planning activities (adapted from Scallan 2003).

ability of required capacity, and attempts to reschedule releases, if necessary (Hopp and Spearman 2011).

The function of manufacturing scheduling is mainly responsible of the resources required for the implementation of the process plan. In addition to the output of the MRP system, the primary inputs utilized by the scheduling function are the outputs derived from the process planning system, in particular, the routing sheets.

The role of the scheduling function is to prioritize the manufacturing orders and specify the timing and sequence of production based on the assignment of jobs to workstations. Thus, the 'when' and 'how many' decisions of production are the main focus of manufacturing scheduling. Subsequently, the schedule is transmitted to the production department for execution. Figure 1.15 illustrates the relationships in terms of inputs and outputs between these functions.

1.4.1 *Lot-sizing*

Lot-sizing is a problem that usually appears in MRP systems. It addresses the fundamental trade-off between maintaining a high number of smaller jobs, which often result in increased setup costs (including materials, tracking expenses, and labour), and/or reduced capacity, versus the alternative of having a smaller number of larger jobs, which can lead to higher inventory levels. These jobs that sizes need to be determined are called "Lots" (Hopp and Spearman 2011)

The problem in its simplest form can be defined as follows: a product is to be manufactured on a given machine over a number of periods. In each period, a demand for the product must be satisfied, and a production cost is incurred with the production of each unit. Additionally, an inventory holding cost is incurred

with the storage of each unit from one period to the next. The objective of the planner is to determine the optimal number of units of the product in each lot to manufacture in each period, with the objective of minimizing the total production cost.
According to Haase, 2012 classifies lot-sizing problems according to:

- **Time scale:** lot-sizing can be carried out either over discrete periods of small size (hours, shifts, days), large size (weeks, months) or very large size (years). Otherwise, a continuous time scale is considered. In our work in chapter 6 we consider discrete large size periods.

- **The planning horizon:** it may be considered to be infinite, finite, or variable. In general, discrete models are based on a finite horizon, while continuous models are based on an infinite horizon. We will consider a problem with a finite planning horizon.

- **Number of products (items):** it can consider the planning of only one product named "Single-item problem", or Multiple products "Multi-item". Lot-sizing problems with single-item are relevant in cases with multiple products, when the products don't share any common resources (Brahimi et al. 2006). Our focus is mainly on problems with a single item.

- **The number of levels (echelon):** is determined by the parent-child relationships between the products. In cases where no such relationship exists, the problem is considered single-level. In contrast, multi-level problems arise when there is a parent-child relationship. The Bill of Materials defines these relationships (Vickery and Markland 1986). In chapter 6 the problem is single-level.

- **Costs:** In addition to the unit production cost, the total cost can include :
  - **Setup costs:** these are the costs incurred each time the preparation of the resource takes place for the production of an item on that resource. In particular, some problems consider setup costs that depend on the sequence in which the products pass through the resource, these are named "sequence-dependent setup costs", and are relevant in the Multi-item case, where it represents the cost of switching the machine from an item to another.
  - **Start-up costs:** represent the cost of switching the machine *on* after it was previously turned *off*. In the absence of such costs, they are considered to be embedded within the setup costs (Karmarkar et al. 1987; Wolsey 1989).
  - **Reservation costs:** Reservation costs are incurred if the resource is kept in a given state (prepared for a product) even if it remains unused for

a certain number of periods. These costs are usually considered along with start-up costs (Karmarkar et al. 1987).

  - **Inventory holding costs:** such costs comprise the expenses incurred in maintaining a product in stock for a specified duration. These costs primarily encompass the cost of locked capital, the risk of loss or obsolescence, taxes, insurance, and the costs associated with maintaining the storage facility.
  - **Capacity-related costs:** these are costs associated with the utilisation of regular capacity and/or supplementary capacity (overtime costs, subcontracting costs or alterations in the workforce).

- **Resource constrains:** If the capacity constraints (e.g. machine capacity, limited number of employees, limits in storage capacity, transport capacity and/or budget) are considered, the problem is defined as "Capacitated". In the absence of such constraints, the problem is said to be "Uncapacitated". The number of machines are also considered as resource-related constraints.
- **Time-consuming tasks:** These include transport time, lead time, preparation time, processing time per unit and production speed. An infinite production speed (or zero processing time per unit of product) is considered in Uncapacitated models. In the opposite case, the capacitated case, a limit is added representing the total time of a period (length), transport, processing and/or setup times are assumed to be zero. Setup times, just like setup costs, may depend on the sequence in which the products pass through the machine.

The introduction of reconfigurable resources to the lot-sizing problem introduces a new set of constraints, times and costs. The costs associated with reconfiguration, which are incurred when changing the state (configuration) of the machine/system to accommodate new capacities and/or functionalities, should be added to the total cost objective function. Furthermore, the reconfigurability of machines introduces an additional layer of complexity to the problem.

1.4.2 *Integrated Process and production planning*

As illustrated in the previous section in Figure 1.15, process planning output serves as an input for multiple production planning activities. In practice, these separated activities are typically conducted in a hierarchical manner, this approach is named "*Offline*" approach, conversely an integrated approach is named "*Online*". Because process planning is conducted without consideration of the status of the shop floor, it will likely result in obtaining a local minimum solution. Furthermore, while the quantity to be produced is of lesser importance to the process planner, it is a significant factor in the selection of manufacturing processes. This is because the

majority of processes and production equipment have an economic batch quantity or a break-even quantity when compared to other processes (Larsen 1993; Scallan 2003).
Additionally, when the manufacturing system is reconfigurable, a notable proportion of the total production cost and time may be attributed to the reconfiguration of the system between successive products. Consequently, optimising the production plan and the corresponding process plans is of paramount importance for achieving meaningful cost and time reductions (Campos Sabioni et al. 2022).
In such a case, a non-optimal (local minimum) solution obtained through conventional process and production planning methods does not meet the required standards. For example, the estimation of reconfiguration costs and times based on sequence-dependent parameters is not applicable in this case, as the reconfiguration effort is dependent on the specific process plans of the involved products. Furthermore, even with fixed process plans, the complexity of setup costs and times, which are influenced by multiple interdependent factors such as part loading and unloading, fixture preparation, and tool changeover, makes precise estimation challenging (Musharavati and Hamouda 2012; S. Zhang and Wong 2016).

It can be seen that the integration of these activities is particularly relevant in the case of RMS, as any excessive reconfiguration or under-utilisation of reconfiguration potential has a negative impact by lowering product quality, increasing costs and extending production time (Khan et al. 2021). This evidence supports the conclusion of X. Gu and Koren, 2018 that intelligent process planning for RMS must integrate production planning decisions.

As a result, the concept of developing an *"ideal"* process plan, representing the best method for manufacturing the product in the conceptual phase and applying it throughout the product's lifecycle (Ishii et al. 1997), becomes inapplicable in a volatile environment and with RMS. Therefore, given the pivotal role of process planning across the product's engineering and manufacturing stages, it should no longer be regarded—and more importantly treated—as a static, independent planning task (Larsen 1993; Phanden et al. 2011).

The integration of process planning and production planning within the context of reconfigurability introduces a new set of challenges. In particular, the representation of interactions between different process plans in the RMS shop floor and the assessment of the influence of these interconnections on the overall performance of the production plan represent critical considerations. Overall, the complexity of RMS introduces an additional layer of difficulty in the optimisation of the already complex problem of integrated process and production planning.

As posited by X. Gu and Koren, 2018, the development of new algorithms is essential to tackle these difficulties. These novel algorithms must be capable of addressing several operational challenges simultaneously, including: (a) the product requirements (such as the urgency and priority of the order); (b) the operations precedence constraints for each product; (c) the machine/station availability and capability; (d) the machine processing time and the travel time on both the forward and return conveyors (See figure 1.5). The algorithms should use this data to determine the usual process planning tasks (mentioned in section 1.3) as well as the optimal order for producing the products, the optimal production routes, taking into account the simultaneous production of multiple products.

1.4.2.1 *Importance of integration*

The tendency for many businesses to adopt a production strategy based on a "one product at a time" approach, rather than considering their entire product portfolio and developing flexible production processes capable of responding to future products and market uncertainties, makes this integration a significant challenge for these businesses (Andersen et al. 2023).
In order to understand the importance of integration, it is first necessary to consider why these activities have traditionally been separated. Process planning and production planning were treated as separate activities for three key reasons:

- **Technological Limitations:** Integration was not feasible due to constraints in computation power and methods employed (ElMaraghy 1993).
- **Market Stability:** Markets were relatively stable, and product portfolios were smaller, especially during the mass production era, making separate planning approaches sufficient.
- **Long Product Life Cycles:** Products had longer life cycles, reducing the frequency of performing process planning (Bortolini et al. 2018).

Nevertheless, there is currently a stronger justification for opting for integrated approaches. The advancement of Industry 4.0 technologies, operations research methodologies and computational capabilities have enabled manufacturers to address the first barrier. Thus, the capacity to better forecast future scenarios, process vast datasets and make optimised decisions within a reasonable time has been enhanced.

As mentioned in section 1.1, market volatility has only grown with globalization and the shift towards mass personalization has even increased this volatility. Modern markets demand RMSs capable of adapting to rapidly changing customer needs and diverse product portfolios. Thus demanding for integrated approaches to plan production within these systems.
Lastly, the reduction in product lifecycles resulted in an increase in the frequency of performing process planning, which consequently became a more of a tactical/-operational activity (Bensmaine et al. 2014a). This facilitated the integration with lower production planning activities.

Furthermore, in the context of an RMS's system design, decisions are inherently embedded within the production and process planning activities, with varying degrees of integration. In the context of process planning with RMS, decisions related to machine, configuration, tool and operation selection, as well as operations sequencing, directly determine the design of the system. The integrated approach to production and process planning also impacts the system design. Finally, although lot sizing is not primarily focused on system design, it specifies the configurations to be used along with lot sizes, and thus has an impact on both the design and the operational aspects of the system.

1.5 CONCLUSION

This chapter has presented the fundamental concepts of RMS, process planning and production planning. An RMS possesses six fundamental characteristics that are integral to its design and functioning. These characteristics serve to distinguish it from conventional manufacturing systems, namely DML and FMS. Furthermore, the chapter concentrated on a specific category of reconfigurable machines, namely RMTs. It also presented an array of classifications for process planning activities and the diverse tasks that are embedded within this long-term planning decision. Lastly, the chapter presented the activities involved in production planning, highlighting the key differences between this and process planning and linking them to RMS. It demonstrated how RMS introduces a new set of challenges to classical production planning activities, necessitating the development of new methods for process planning within RMS and for production planning within RMS. The importance of integrating these activities was also presented, as they are particularly relevant in today's dynamic markets, where RMS is designed to operate.
The following chapter will focus on the operational research aspect of the thesis, presenting the various optimisation techniques employed throughout the subsequent chapters of this thesis.

2

OPTIMIZATION METHODS

This chapter will introduce some fundamental concepts related to optimization methods, both mono-objective and multi-objective, which will be employed later in this thesis. First exact solution methods applicable for mono-objective optimization and their counterparts, the approximate methods, are presented. Subsequently, the chapter will present an overview of multi-objective optimization, including techniques for utilising exact approaches, as well as approximate methods and evaluation metrics for multi-objective problems.

2.1 INTRODUCTION

Before discussing optimization methods, it is important to begin with an introduction to complexity theory in operations research, since understanding how to solve a problem requires knowing how hard this problem is, and thus what's the most appropriate technique to apply. The goal of complexity theory is to categorise problems in order to predict their level of difficulty. Researchers consider that a problem is "*easy*" if an algorithm can solve it, to optimality, and its computational complexity is bounded by a function that is polynomially dependent on the amount of data to be treated.

To assign a problem to a complexity class, we examine the performance of the best algorithm — which may or may not be known — able to solve it to optimality rather than testing all possible algorithms (Taillard, 2023). The basic complexity classes are:

- **P (Polynomial)**: includes problems that can be solved to optimality within a time that grows polynomially with the size of the input. Accordingly, a problem is classified as belonging to difficulty class P if an algorithm exists that can solve it in polynomial time, even if it is not the best one. This category of problems is considered to be the easiest to solve compared to other classes.

- **NP (Non-Deterministic Polynomial)**: problems in this class may not be solvable to optimality in polynomial time, However, the correctness of a given solution can be verified in a polynomial time.

- **NP-Hard (Non-Deterministic Polynomial Hard)**: these are harder problems where no polynomial-time algorithm is known to solve them.
- **NP-Complete (Non-Deterministic Polynomial Complete)**: these problems belong to the class NP and are also NP-Hard. No NP-complete problem that can be solved in polynomial time is known.

Other classes of complexity exist such as: **P-SPACE** which are problems that can be solved by a computer whose memory is limited by a polynomial data size. **Class L** are problems that can be solved by a computer whose working memory is constrained by a polynomial in terms of amount of data, and ignoring the space required for the storage of the problem. **Class NC** comprises problems that can be solved in poly-logarithmic time on a computer that has a polynomial number of processors. It follows that the time required to address these problems in parallel is less than the time required to read the data sequentially. The sorting of the elements of an array fall within this category (Taillard, 2023).

2.2 MONO-OBJECTIVE OPTIMIZATION

Optimization is the process of finding the best values of a set of variables that correspond to the maximum or minimum of one or more objective functions. Optimization has many applications in engineering, science, business, economics, etc., using mathematical models and algorithmic methods. Without optimization of design, operational, production and engineering activities will not be as efficient as they are today (Bensmaine, 2013).

It is essential to formally express or model the fundamental aspects of the problem in order to ensure that it has been fully understood and will be correctly solved. There are a number of ways in which a problem can be represented.
The two main elements that must be defined in order to solve a problem are the *objective function* ($f(x)$) and the set of *constraints* ($g_i(x)$ and $h_i(x)$). An objective function represents the goal of the optimisation process, which is a function that has to be either maximised or minimised. Constraints, on the other hand, represent a set of conditions that must be satisfied. While x is the decision variable, and m,p are the numbers of inequality and equality constraints, respectively.

$$\begin{aligned} \text{Minimize (or Maximize):} \quad & f(x) \\ \text{Subject to:} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(x) = 0, \quad j = 1, \ldots, p, \end{aligned}$$

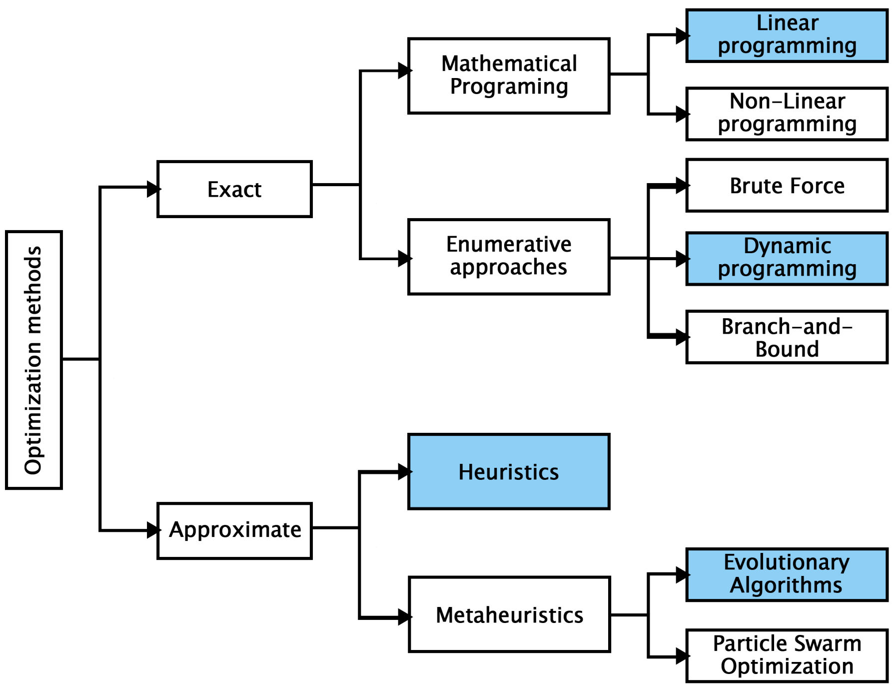


Figure 2.1: Classification of optimization methods with methods employed in this thesis in blue.

Figure 2.1 illustrates a classification of optimization methods where the methods in blue are the ones that will be used in the next chapters to solve presented optimization problems. From the figure we can distinguish two main types of optimization methods: Exact and approximate methods.

2.3 EXACT METHODS

The usage of exact methods is a common approach to solve polynomial problems[1]. These methods are designed to guarantee the optimality of the solution obtained, provided that sufficient time is allowed for the optimization process to complete.

2.3.1 *Mathematical programming*

In these approaches, a mathematical model for the problem is developed and then employed in a solver[2] for optimization. A mathematical model is defined as a formal mathematical representation of a problem, constructed by translating the real-world scenario into a set of mathematical expressions that capture the rela-

1 Exact solution approaches do not guarantee that P problems are solved in a polynomial time.
2 A solver is a software used to find the optimal solution to a mathematical model, example of such software are CPLEX, Gurobi, GAMS,...

tionships between different factors influencing the decision-making process.
The model is composed of variables representing various elements of the problem, where the values of some of these variables needs to be determined through optimization.
Two main sets of variables constitute the core components of the model. The first set is referred to as the *decision variables*, which represent the decisions to be made within the problem. These variables directly influence the outcome of the solution and are subject to constraints, for example, the number of units to be produced of a given product or the amount of resources to be allocated to a particular task. The second set is known as *parameters*, which are fixed quantities or constants within the model that describe elements of the problem that do not change during the optimization process. These parameters often include factors such as production costs, resource availability, or time constraints. Together, decision variables and parameters form the foundation of the mathematical model, enabling the solver to explore different combinations of decisions while adhering to the problem's constraints and objective function.

2.3.1.1 *Integer Linear Programming*

In these models, all decision variables are integers and the objective function and all constraints are linear. Conversely, Mixed-Integer linear programming (MILP) contains continuos and integer decision variable and 0-1 Linear Program (01-LP) contain boolean variables only.
01-LP, ILP, and MILP are widely used for modelling problems in production planning and process planning. These approaches are used in algorithmic design paradigms for solving combinatorial optimization problems, particularly with methods like Branch-and-Bound and Branch-and-Cut (Taillard, 2023).

2.3.1.2 *Non-Linear Programming (NLP)*

As their name indicates, these problems are characterised by the presence of a non-linear objective function or constraints. A special case of this problem is quadratic programming, which is defined by the a quadratic objective function and linear constraints. Solving these problems is generally more challenging than linear problems due to the potential for multiple local minima or maxima and the need for specialized algorithms. They arise in many real-world applications, such as engineering design, economics, and machine learning, where the relationships between decision variables are inherently non-linear.

2.3.2 *Brute Force*

Brute Force is a straightforward problem-solving approach that systematically enumerates all possible solutions and evaluates all of them to find the optimal solution. While simple and easy to implement, it is computationally expensive for large problems, as its time complexity grows exponentially with the problem size. This method is often used for small sized problems or as a benchmark to compare the performance of more sophisticated algorithms or as a last resort when other approaches fail. Despite its inefficiency, brute force is an exact solution approach, thus guarantees finding the optimal solution, making it valuable for small-scale problems or scenarios where precision is critical.

2.3.3 *Dynamic Programming*

Dynamic programming is a solution method introduced by Bellman, 1954, commonly defined as breaking down a problem into smaller, easier-to-solve subproblems, each sub-problem is solved optimally in an iterative manner, then the solution to the global problem is constructed (Taha and Taha, 2003).
While this definition is correct, it can be somehow misleading. Since the definition might be understood as that dynamic programming is a decomposition heuristic such as divide-and-Conquer, which is not.
An important aspect of dynamic programming, which is surprisingly overlooked when discussing it, is that it is applicable only to problems that exhibit certain structures. These problems are named by Bellman, 1954 "*Multi-stage decision processes*", to understand what it means, one needs first to understand two terms:

- **Stage**: is a level at which a decision is made (for example a moment in time), which is composed of states.
- **States**: are possible situations in which the system can exist.

In a multi-stage decision process, decisions are made in each stage which "transform" the system from a state to another in the next stage. Thus the state at which we exist in a certain stage is dependent on the previous decisions made in previous stages. This overlapping nature of subproblems (stages) can be represented through a recursive function. It is easy to see that this structure is not always existing in an optimization problem.
This leads us to the main idea behind dynamic programming: is that instead of looking for the *optimal solution* the decision maker looks for an *optimal policy*. Bellman defines a policy as: "*a rule for telling what decision to make in terms of the current position of the system*" Bellman, 1966. Thus dynamic programming consists of a tool to make a decision in each stage depending on the current and future states, to

minimize a certain function.
What we try to highlight here is that the difficulty is not only in finding the optimal policy but also on how to identify a problem's stages and states and how apply the optimal policy and store the results of these subproblems to avoid redundant computation, and it should be mentioned that not every problem's stages are clear and easy to spot.

Dynamic programming have proved to be a very effective solution method, nevertheless, as with any exact solution approach, the number of stages and states that must be enumerated increases exponentially with the size of the problem. This phenomenon, which Bellman has termed the "*curse of dimensionality*," renders dynamic programming very hard to apply for problems of considerable scale. Even though the approach can be very effective when compared to other exact methods such as Linear programming, as will be demonstrated in chapter 6.

2.4 APPROXIMATE METHODS

Approximate approaches represent a natural extension to exact solution approaches, offering an improvement in computational efficiency. In some cases, decision-makers are willing to accept a certain degree of compromise in terms of solution quality, provided that the solution is obtained in a relatively short computational time. Therefore, although approximate methods do not guarantee the optimal solution, they allow for a certain degree of flexibility in balancing solution quality and computational time.

2.4.1 *Heuristics*

Heuristics are problem-specific, these algorithms are designed to address a specific problem or instance, and thus may not be applicable to other problems. They can be categorized into construction and improvement heuristics. Construction heuristics creates a solution from scratch without using an initial solution. On the other hand, improvement heuristics are algorithms that use an initial solution and through the iterations it improves the quality of the solution, if possible (Brahimi, 2004).

2.4.2 *Metaheuristics*

While heuristics are specific, metaheuristics are general-purpose algorithms that can be designed to solve a large range of solutions approaches. They represent high-level frameworks for guiding the development of heuristics for solving a problem (Talbi, 2009). One of the most widely employed metaheuristics is Genetic Algorithm (GA), which belongs to the evolutionary algorithms family.

2.4.2.1 *GA*

Genetic algorithms were developed by Holland, 1973, who drew inspiration from the natural process of evolution, particularly the operations on chromosomes and genes. GA is population based, thus a group of solutions, named a generation or population, is considered in an iteration instead of only one element. This population represents a group of solutions, named chromosomes. The algorithm is composed of three main genetic operators: selection, crossover, and mutation.

- Selection: Consists of selecting of parents for crossover and mutation where these chromosomes will be used to generate the new population, also named "offspring". Selection can be random or tournament based. The main focus is on selecting high quality parents, since based on the elitism principle, high quality parents tend to give high quality child chromosomes. The choice of a selection strategy depend on the problem at hand.
- Crossover: Consists of combining the chromosomes from the two parents to give offsprings, usually by cutting the parents' chromosomes into two halves and combining each parent's part with the other one's to generate two new child solutions.
- Mutation: The smallest unit of a chromosome is known as a gene. Consequently, the mutation operator modifies a gene of the child chromosome randomly, thus preventing the replication of identical solutions present in the parent population.

2.5 MULTI-OBJECTIVE OPTIMIZATION

In practical situations, a decision-maker is interested in optimising a number of objectives simultaneously, rather than just one as in the mono-objective case. This may, for example, involve minimising production time while minimising production cost. In this case, mono-objective optimisation methods, when applied in their mono-objective form, are not applicable since they are based on a comparison of solutions evaluated according to a single criterion, resulting in the identification of a single "best"—Or "optimal" if we are employing an exact solution approach—(minimal or maximal) solution among the compared ones.
Thus a new class of optimization have emerged named *Multi-objective Optimization*. Where the optimization problem with two objectives f_1 and f_2 takes the following structure:

$$\begin{aligned}
\text{Minimize (or maximize):} \quad & f_1(x), f_2(x) \\
\text{Subject to:} \quad & g_i(x) \leq 0, \qquad i = 1, \ldots, m, \\
& h_j(x) = 0, \qquad j = 1, \ldots, p,
\end{aligned}$$

Thus multi-objective optimization methods return the "best"—Or "optimal"— (minimal or maximal) set of solutions instead of only one solution.

In the context of mono-objective optimisation, a solution A is said to be "better" than a solution B if, in a minimisation problem, $f(A) < f(B)$. However, in the context of multi-objective optimisation, where multiple objectives are considered, this is replaced with the concept of dominance. In this case, solution A is said to "dominate" solution B if A improves at least one objective over B without deteriorating at least another objective.

It can be observed that multi-objective optimisation methods result in the generation of a set of non-dominated solutions, wherein none of the solutions are dominated by any other known solution.

A solution that is not dominated by *any* other feasible solution is defined as a "Pareto optimal solution". A set of Pareto optimal solutions for a given problem is referred to as the Pareto front[3].

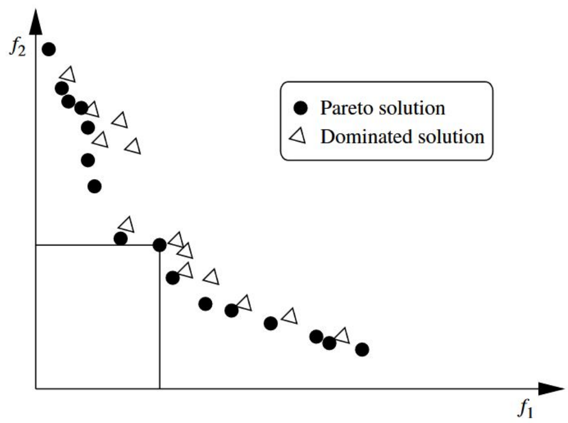


Figure 2.2: Representation of Pareto front and dominated solutions for a case with two objectves (Talbi 2009).

2.5.1 *Harmony and Conflict*

While multi-objective optimization deals with problems that have multiple objective functions, not all problems with multiple objectives are considered as multi-objective optimization. In certain cases the objectives are dependent where minim-

3 Sometimes referred to as the "True Pareto front".

izing one would directly minimize the others. For example in a problem with the objectives of minimizing the energy cost and minimizing energy consumed, since energy cost is computed as energy consumed multiplied by the unit energy price, minimizing one of these objectives will automatically minimize the other,thus this problem can be converted into a mono-objective optimization problem.
In such a case, where only one optimal solution exists in the Pareto front, the objectives are said to be in "Total harmony" (Chugh et al., 2023), and mono-objective optimization methods should be used.
Conversely, two objectives are said to be in "conflict", if there's at least two *different* solutions in the true Pareto front. There are multiple metrics used to quantify the level of conflict between objectives (See Chugh et al., 2023).

2.5.2 *Scalarization techniques*

Once we have established that we have conflicting objectives in a problem. There are multiple approaches for solving the problem. We are interested in two main approaches relevant to our thesis. First consists of solving multiple sub-problems where each one yield a non-dominated solution. Second, are population based methods where a set of solutions are considered at a time and updated in every iteration.
In the first type of approaches, the multi-objective problem is converted into mono-objective problems using a "Scalarization technique" (Talbi 2009).

2.5.2.1 *Weighted Sum*

In the weighted sum technique, a weight w_i is assigned to each objective[4] and the sum of all objectives is minimized. By varying the weights of objectives the user can explore different zones of the Pareto front. Usually $\sum_i w_i = 1$. The problem becomes:

$$\begin{aligned} \text{Minimize (or maximize):} \quad & w_1 f_1(x), w_2 f_2(x) \\ \text{Subject to:} \quad & g_i(x) \leq 0, \qquad i = 1, \dots, m, \\ & h_j(x) = 0, \qquad j = 1, \dots, p, \end{aligned}$$

2.5.2.2 *ϵ-Constraint*

In the ϵ-constraint technique, the problem consists of minimizing only one objective, subject to constraints on the other objectives. A constraint limit is defined to

4 Usually the weights are defined by decision maker.

limit the values of other objectives, and are set to control the zones of the objective space searched. The problem structure becomes:

$$\begin{aligned} \text{Minimize (or maximize):} \quad & f_1(x) \\ \text{Subject to:} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(x) = 0, \quad j = 1, \ldots, p, \\ & f_2(x) \leq \epsilon \end{aligned}$$

2.5.2.3 *Normal-Boundary Intersection*

Introduced by Das and Dennis, 1998 the Normal-Boundary Intersection method demonstrated its independence from the relative scales of the objective functions, and proved to be efficient in generating an evenly distributed set of points in the Pareto set when provided with an evenly distributed set of parameters. This is a notable advantage compared to the popular weighted sum method, which lacks this property (Das and Dennis 1998). Furthermore, the NBI method allows for an effective solution search control by adjusting the β vector, which is easier than ϵ-constraint bounds especially when the Pareto front shape is not known. By varying β values we can explore different regions of the search space. This attribute makes it easy to implement and also to control the search process.

The method comprises three key phases: i) Initially, the ideal point is determined by minimizing each objective individually. ii) Subsequently, the ideal point is utilized to construct the payoff matrix Φ. iii) Lastly, a new subproblem is formulated and a new real variable t is introduced. The new objective becomes the maximization of t.

This objective function and constraints represent the typical structure of NBI subproblems (for more details, see Vahidinasab and Jadid, 2010). These subproblems are solved for various values of β_1 and β_2 with $\hat{n}$ being the unit normal vector.

$$\begin{aligned} \text{maximize:} \quad & t \\ \text{Subject to:} \quad & g_i(x) \leq 0, && i = 1, \ldots, m, \\ & h_j(x) = 0, && j = 1, \ldots, p, \\ & \Phi\beta_1 + t\hat{n}_1 = f_1(x) \\ & \Phi\beta_2 + t\hat{n}_2 = f_2(x) \end{aligned}$$

2.5.3 *Multi-objective Metaheuristics*

As with the mono-objective case, metaheuristics are employed as approximation methods in multi-objective optimisation. The resulting solutions comprise a set of non-dominated solutions, which is referred to as a "Pareto front approximation".

2.5.3.1 *NSGA-II*

The most well-known extension of the GA for multi-objective optimisation is NSGA-II, developed by Deb et al., 2002, which has been extensively employed in the domain of multi-objective optimisation. It has demonstrated particular effectiveness in solving process planning with RMS problems(See chapter 3).
The method comprises a non-dominated sorting and crowding distance mechanism, which produces a set of non-dominated solutions, starting from an initial population. The specific steps and details of NSGA-II can be found in Algorithm 1, with the population size denoted by $Population_{size}$, the crossover probability by $P_{crossover}$, and the mutation ratio by $R_{mutation}$.

```
Data: Population_size, P_crossover, R_mutation, limit_time
Result: Non-dominated solutions
ParentPopulation ← Initial population;
while CPU_time < limit_time do
    Generate childPopulation from ParentPopulation (P_crossover, R_mutation) ;
    population = ParentPopulation ∪ childPopulation;
    F = fastNonDominatedSorting (population);
    i = 0;
    while |newPopulation| + |F_i| ≤ Population_size do
        Add F_i to newPopulation;
        i = i + 1;
    end
    F_i = crowdingDistanceSorting(F_i);
    while |newPopulation| < Population_size do
        Add an element from F_i to newPopulation;
    end
    ParentPopulation = newPopulation;
    Update CPU_time;
end
```

Algorithm 1: NSGA-II

The two essential parts of NSGA-II are:

- Fast Non-dominated sorting: In this operation the solutions of a given population are grouped into "fronts". Between solutions belonging to the same front, no solution is strictly better than any other solution of the same front. This operation takes the majority of total computational time of NSGA-II (Fang et al., 2008).
- Crowding distance: After solutions are ranked in fronts, solutions of a front undergo another ranking which is named crowding distance. The crowding distance is defined as the circumference of the rectangle defined by its left and right neighbours, and infinity if there is no neighbour.

2.5.3.2 *MOEA/D*

MOEA/D, introduced by Q. Zhang and Li, 2007, has emerged as a powerful competitor to other evolutionary algorithms, such as NSGA-II (H. Li and Zhang, 2008; Q. Zhang and Li, 2007). This metaheuristic operates by decomposing the multi-objective problem into single-objective subproblems, with different weights assigned to the objectives for each solution in the population. The objective is to simultaneously optimise multiple conflicting objectives, thereby generating a diverse set of non-dominated solutions that capture the trade-offs between objectives.

A variety of decomposition approaches have been proposed, including the weighted sum, Tchebycheff, and boundary intersection (for further details, please refer to Q. Zhang and Li, 2007). The Tchebycheff decomposition is employed, whereby solutions are compared based on the maximum normalised difference relative to a reference point. The specific algorithmic details of MOEA/D are illustrated in Algorithm 2.

```
Data: Population_size, P_crossover, R_mutation, T, limit_time
Result: Non-dominated solutions
ParentPopulation ← Initialize population;
Determine the reference point
  z_i = min{f_i(x^1), f_i(x^2), ..., f_i(x^N)},   i = 1, ..., m;
Generate the random weight vectors w^1, ..., w^N;
Compute the Euclidean distances between every two weight vectors and
  then determine the T closest weight vectors to each weight vector;
while CPU_time < limit_time do
    Generate childPopulation from ParentPopulation (P_crossover, R_mutation, T) ;
    Update the reference point z_i;
    Update ParentPopulation with the solutions from childPopulation for
      neighbor subproblems based on the Tchebycheff approach;
    Update non-dominated solutions F_0 = NonDominatedSorting
      (childPopulation);
    Update CPU_time;
end
```

Algorithm 2: MOEA/D algorithm

2.6 PERFORMANCE INDICATORS

When comparing two Pareto front approximations, identifying the best front might not be an easy task, such as the example of two fronts presented in figure 2.3. Thus performance indicators are used to evaluate the quality of the different obtained Pareto fronts approximations. According to Audet et al., 2021 there are four categories of multi-objective performance indicators: Cardinality indicators, Conver-

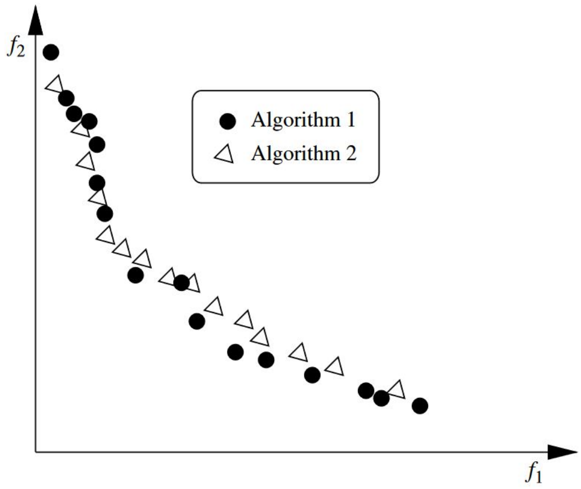


Figure 2.3: Representation of two Pareto front approximations for the case with two objectives (Talbi 2009).

gence indicators, Distribution and spread indicators, and indicators that combine Convergence and distribution.

2.6.1 *Cardinality indicators*

Cardinality indicators are used to quantify the number of non-dominated points generated by an algorithm.

- Number of Solutions (NS): represents the number of solutions found ($|A|$), a higher value of NS means more solutions found, regardless of the quality or the spread of these solutions.

- **C-metric**: denoted as C(A,B), is a measure used to assess the dominance relationship between the two sets of solutions A and B (He et al. 2022; P. Li et al. 2023). It calculates the proportion of solutions in set B that are dominated by at least one element within set A. Its value ranges from 0 to 1 (inclusive) where:
  - A value of C(A, B) equal to 0 implies that no solution in B is dominated by any solution in A (ideal scenario for set B meaning that B completely dominates A).
  - A value of 1 signifies that all solutions in B are dominated by solutions in A (worst case for set B).

  It should be noted that C(A, B) does not necessarily equal C(B, A). The C-metric is calculated using the following formula:

$$\text{C(A,B)} = \frac{|\{X_2 \in B | \exists X_1 \in A, \quad X_1 \text{dominates} X_2\}|}{|B|}$$

2.6.2 *Convergence indicators*

These indicators measure how close a set of non-dominated points is from the true Pareto front in the objective space.

- **Generational distance (I_{GD}):** computes the average distance between the points in the obtained Pareto front approximation and their nearest point from a reference set. The reference set is generally represented by the true Pareto front.

2.6.3 *Distribution and spread indicators*

These indicators quantify the distribution of a Pareto front approximation. Focusing mainly on the aspect that points should be far away from each other, and that they are uniformaly distributed.

- Spacing metric (SM): reflects the diversity of solutions within an obtained front. A lower SM value indicates a more even distribution of solutions across the front, suggesting good diversity (Khezri et al. 2021; P. Li et al. 2023). SM is calculated using the following formula:

$$\mathrm{SM} = \sqrt{\frac{\sum_{i=0}^{|A|} (d_i - \bar{d})^2}{|A|}}$$

  Where d_i is the normalized rectilinear distance between the point i and the closest point to it, and $\bar{d}$ is the mean value of all distances.

2.6.4 *Convergence and distribution indicators*

Capture both the properties of convergence and distribution.

- Hypervolume (HV): represents the objective space covered between the solution points and a reference point Z_{ref}. In our case, we set Z_{ref} as Z_i^{max}, representing the maximum value of objective i from the reference set (Talbi 2009). Figure 2.4 illustrates the space represented by the HV indicator as well as a reference point Z_{ref}.

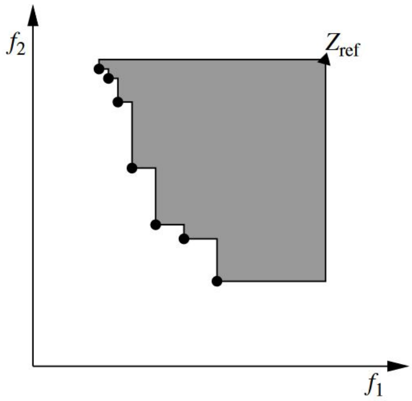


Figure 2.4: Hypervolume indicator for Pareto front approximation. (Talbi 2009)

2.7 CONCLUSION

This chapter presented an overview of the subject of optimisation methods. We began by introducing the concept of optimization in general, with a particular emphasis on mono-objective optimization, which consists of a single objective. This field can be divided into two main categories of solution methods: exact and approximate. Approaches that guarantee optimality of the solution generated are classified as exact solution methods. In contrast, approximate methods are faster but do not guarantee optimality. We have discussed some exact solution approaches, placing particular focus on linear and dynamic programming. In the case of approximate methods, we have introduced the concepts of both heuristics and metaheuristics.

Conversely, multi-objective optimisation is concerned with cases where more than one objective must be optimised. We have presented a number of Scalarization techniques, with a particular focus on the NBI strategy. Subsequently, two evolutionary multi-objective optimisation metaheuristics were presented: NSGA-II and MOEA/D. Ultimately, the quality of the obtained solutions is evaluated through the use of performance indicators. The four categories are introduced, namely cardinality, convergence, spread, and both convergence and spread-based indicators. The section gives particular emphasis on those employed in this thesis.

The following chapter presents a review of the literature on production planning problems for RMS, with a particular focus on the optimisation methods employed to solve these problems.

3

AN ADAPTED LITERATURE REVIEW ON PRODUCTION PLANNING AND RMS LITERATURE

This chapter presents a literature review on production planning, with an emphasis on process planning and lot-sizing, it presents also the related production planning literature on RMS. It begins with a broad overview of production planning and lot-sizing activities. It then explores the process planning function in depth before moving on to RMS systems and RMS-related literature reviews. It concludes the chapter by presenting RMS related papers for process planning, production planning, and the integration of both.

3.1 PRODUCTION PLANNING

As outlined in chapter 1, production planning is an umbrella term encompassing a range of activities related to the planning of the execution of production processes. Given the focus of this thesis on lot-sizing and, to a lesser extent, scheduling, we will primarily focus on the first activity's literature in this section.

3.1.1 *Lot-sizing*

As previously stated, lot-sizing can be classified into two categories: single-item and multi-item. The problem addressed in chapter 6 is a single-item one.
Despite being a special case of the multi-item problem and "easy" to solve (polynomial), the Single-item Lot-sizing problem (SILSP) has nevertheless attracted considerable interest from researchers, with Brahimi et al., 2006 conducting a literature review on the subject, which they updated later in Brahimi et al., 2017.
This interest is primarily due to the fact that SILSP models certain tactical production and distribution problems. Furthermore, SILSP arises as a sub-problem in a number of lot-sizing problems, including multi-item lot-sizing (Brahimi et al. 2017).

The main focus in this thesis is on deterministic, finite planning horizon SILSP, which are solved mostly in a Mono-objective context, focusing on minimizing the total cost.
These problems consist of a finite number T of periods during which production takes place, and in each period a known demand quantity of the item has to be satisfied. The production plan (i.e., lot sizes) has to fulfil this demand while minimizing the total cost, usually composed of production costs p_t, holding costs h_t, and setup costs s_t with t being the index of periods.

Lot-sizing problems can be further classified depending on the periods length to, small and big-bucket problems. Small bucket problems have short production periods, usually used to model multi-item problems, allowing only a production of one item or setup of the machine in a given period. The extension to a more generalized model of the problem is big bucket models, where periods are longer so that a period can have production of multiple products and/or setups (Quadt and Kuhn 2008).

The pioneering work in the field of SILSP was conducted by Wagner and Whitin, 1958. Their seminal paper studied the Uncapacitated SILSP proposing a DP algorithm that runs in $O(T^2)$ and in $O(T)$ in the case with *no speculative motives*[1].
The DP is developed based on two key properties:

- **Property 1:** Production is only possible in a time period t in which the inventory level entering that period is zero. Thus, the quantity produced in a period (x_t) is equal to zero if the inventory level (I_{t-1}) is non-zero.
- **Property 2:** When production occurs in a period t, the quantity manufactured is equal to the demand of an integer number of future periods of t (i,e., $x_t = \sum_{i=t}^{t+n} d_i$ with $n \in \mathbb{Z}$).

Later on, the DP of W-W have been used to solve multiple extensions of SILSP. Zangwill, 1966 generalized the DP to solve the problem with backlogging. When backlogging is allowed in a problem the demand for a period can be satisfied from inventory of a future period with a certain penalty cost named backlogging cost b_t.
On the contrary, another alternative is the case with lost sales, in this extension the demand that isn't satisfied in its period will be lost. Aksen et al., 2003 extended W-WDP, using W-W properties to develop a $O(T^2)$ DP for the SILSP with lost sales.

When capacity constraints are considered, the problem is said to be "*Capacitated*" (See chapter 1, section **??**). Capacitated SILSPs are harder to solve than their Uncapacitated counterparts. However some cases are proven to be polynomially solvable depending on the structure of the production, setup, and holding costs as well as the structure of the capacity limit (Brahimi et al. 2006).

1 *no speculative motives* is the case where $p_{t-1} + h_{t-1} \geq p_t \quad \forall t \in T$.

In some real world planning problems it is not sufficient to only consider capacity limits on production, but also on inventory level, Love, 1973 was the first to model this problem and solved it using an $O(T^3)$ DP which was later improved to $O(T^2)$ by Toczylowski, 1995.

In contrast to the general case where x_t can take any integer value, some production systems manufacture products by batches, these problems are modelled with constant batch sizes or step-wise production constraints on lot sizes. In such a case, production is conducted in fixed-size batches, with the decision being on the number of batches to produce in a given period.
Such a work can be found in Van Vyve, 2007 who proposed an $O(T^3)$ algorithm for SILSP with batch sizes and backlogging.

As mentioned in chapter 1, the objective of a lot-sizing problem is to find an optimal balance between two contrasting approaches: a high number of setups with low inventory carried, and a low number of setups with high inventory carried.
However, in certain cases, it is challenging to accurately determine the precise (or best) setup cost. It is more practical to estimate a minimum batch size (referred to in lot-sizing literature as Minimum Order Quantity (MOQ)) below which it would not be cost-effective to produce (Anderson and Cheah, 1993).
This represents another class of lot-sizes related constraint, in such a case, if $x_t > 0$ then production must surpass a certain minimum value L_t, ($x_t \geq L_t$).
Anderson and Cheah, 1993 was the first to model the problem, they developed a pseudo polynomial DP for solving the Uncapacitated SILSP with fixed MOQ (not time variant), they also employed the DP in a Lagrangian relaxation heuristic to solve the multi-item version.
Later, Okhrin and Richter, 2011a developed an $O(T^2)$ DP for solving the problem with constant production and holding costs and fixed MOQ, whereas they developed an $O(T^3)$ DP in Okhrin and Richter, 2011b for the extension of the problem, adding fixed capacity limit.
As an extension to the work of Okhrin and Richter, 2011b, Hellion et al., 2012 proposed an $O(T^6)$ algorithm for the problem with concave production and storage costs.
For the case with time-varying MOQ, Park and Klabjan, 2015 considered non-increasing linear costs and non-increasing L_t, proving that it is a polynomial case, and providing a polynomial algorithm.
Absi et al., 2016 studied the complexity of the Uncapacitated SILSP with MOQ and proved that it is NP-Hard in the general case of MOQ (general time-variant).
Table 3.1 is retrieved from Park and Klabjan, 2015 which classifies MOQ related lot-sizing papers.

Other works consider changes to cost function structures, with a particular focus on production and holding costs. In Wagner and Whitin, 1958, a fixed holding

Table 3.1: Polynomial cases of lot-sizing with MOQ as classified by Park and Klabjan, 2015.

	Constant capacity	Uncapacitated
Fixed MOQ	Hellion et al., 2012 Okhrin and Richter, 2011b	Okhrin and Richter, 2011a Anderson and Cheah, 1993
Time varying MOQ	Park and Klabjan, 2015	L. Li et al., 2011

cost (h_t) is incurred with each unit stored. Thus, the holding cost function is linear, taking the form $h_t I_t$. Furthermore, the production cost is given by the expression $p_t x_t + s_t \delta(x_t)$, with $\delta(x_t) = 1$ if $x_t > 0$ and zero otherwise. However, other cost function structures exist, see Brahimi et al., 2017 for more details.

Additionally other costs can be added other than for production, setup and storage, such as startup costs. In SILSP a startup cost is incurred in a period where $x_t > 0$ and $x_{t-1} = 0$. These costs represent the cost of restarting the machine after it was turned off previously. Karmarkar et al., 1987 was among the first to study this problem, they proposed a DP as an extension of W-W that runs in $O(T^2)$, and a Lagrangian relaxation for the capacitated case.

Independently, Wolsey, 1989 proposed valid inequalities and a polynomial separation algorithm for solving the problem.

In the majority of lot-sizing works that consider startups, a reservation cost or a setup cost is included in the objective function Karmarkar et al., 1987; Wolsey, 1989, such costs can be used to minimise the number of setups or reservations, though even without startup consideration setup costs serve the same purpose.

3.1.1.1 *Changeability in lot-sizing*

Since our main focus is on *reconfigurability*, we have a particular interest in investigating changeability consideration in lot-sizing classical works.

A similar concept to that of changeability in lot-sizing can be found in manufacturing and remanufacturing-related works, whereby the system can be adjusted from the manufacturing state to the remanufacturing state. While these works are not entirely aligned with the RMS concept, they do exhibit certain similarities that are worthy of examination. For example, Giglio et al., 2017 proposed a formulation for the integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing and remanufacturing systems, where the processing speed to a product can be adjusted, and the scheduling is mainly considered with sequencing manufacturing and remanufacturing.

Another class of lot-sizing works that consider changeability in the system are capacity adjustment works. In these problems the capacities of periods are adjustable from period to period. For example, Ou and Feng, 2019 developed a model that optimizes production capacity decisions over multiple periods. Their

model considers time-varying demand, capacity acquisition and adjustment costs, demonstrating how flexible capacity adjustments can reduce overall costs.
Although numerous assumptions explored in the field of lot-sizing share similarities with those of RMS, they nevertheless fail to adequately address the issue of reconfigurability of machines and systems.

3.2 PROCESS PLANNING

It is a commonly held view that the plans created by two human process planners for the same part will rarely, if ever, be identical. The variability observed in manually generated plans, particularly when the objective is to achieve optimal efficiency or specific criteria, underscores the significance of CAPP as a tool to address these challenges (Chryssolouris 2013).
In their detailed review of CAPP, Xu et al., 2011 identified several key problems types treated in CAPP literature, categorizing them as follows: *tool selection, setup planning, selection and sequencing of operations, decision-making models, integration of process planning with production systems*, and *energy-conscious and energy-efficient CAPP considerations*.

As stated in chapter 1, CAPP can be categorized into generative and variant. *Generative CAPP* systems are harder to develop. Still, some authors attempted to develop generative CAPP models, such as Sadaiah et al., 2002 who developed a CAPP for prismatic components. *Variant CAPP*, or retrieval process planning, on the other hand, is relatively faster fore generating process plans however of lesser quality.

One of the recent papers on process planning is Luo et al., 2022, who differs from the normal process planning works, proposed a 0-1 mathematical programming formulation to solve the flexible process planning problem where both sequencing of operations and operations selection are performed. They considered tools and TADs-related constraints as well as precedence constraints between operations.

3.2.1 *Integrated Process and Production planning*

As process planning is mainly focused on "how" and "how long" to manufacture a product, it considers only one product at a time. Thus, when considering multiple products in process planning, we naturally integrate decisions from production planning activities, mainly "when" and/or "how many" questions (see Chapter 1). Integrated Process Planning and Scheduling (IPPS) is among the most studied problems in this literature, where the "when" to perform different operations question is integrated with process planning decisions. IPPS has been considered in S. Zhang and Wong, 2016, who studied the impact of varying different sequence-dependent times, such as tool change and fixture preparation times, on the makespan.

3.3 RMS

This section presents an overview of the existing literature reviews on RMS, and positions our thesis projects within their different classifications.

Multiple literature reviews have been carried out on RMS-related papers, covering various topics, including design methods for RMS, research and development of RMTs, RMS design, management and industrial application, and sustainability aspect in RMS. Andersen et al., 2015; Bortolini et al., 2018; Koren et al., 2018; Skärin et al., 2022. The Yelles-Chaouche et al., 2021 review focused on optimization problems in RMS, with a sub-section dedicated to process planning problems.
The review by Andersen et al., 2015 begins by detailing their methodology, including how they searched, retrieved, excluded, and categorized the reviewed papers. Their classification framework is structured around six organizational levels proposed in the literature. At the network level, strategic decisions, such as selecting partners, are addressed. The factory level focuses on the factory layout aspects. The segment level, closely related to the factory level, encompasses all activities required to manufacture and prepare products. Moving down, the system level considers the manufacturing system, covering elements like process planning. Within this system, the cell level focuses on subsystems, such as cell design, while the workstation level addresses individual machines, including the design of RMTs.
The work presented in chapter 4 can be considered as system level, with the consideration of factors that aren't directly related to manufacturing (such as factory layout), it can also reach the segment level, which makes it, according to the authors, an interesting contribution since it differs from the classical "conceptual" works in factory level to a more of a "functional" work.
In another study, Gadalla and Xue, 2017 explore the latest advancements and challenges in the development of RMTs, categorizing research into three main areas. Architecture design often considers the choice between semi-open and open modular approaches, allowing for the addition or removal of machine modules. Configuration design and optimization involves evaluating and optimizing configurations and reconfiguration processes to identify the most effective designs. Lastly, system integration and control focuses on designing interfaces and Open Architecture Controllers to enable the seamless transfer of motion, energy, and data between different modules.
Koren et al., 2018 begin their discussion by introducing key concepts such as RMTs, Reconfigurable Inspection machines, and RMS. Building on the six core character-

istics of RMS, they outline six fundamental principles for designing such systems. These principles emphasize the importance of cost-effective scalability to adapt manufacturing system capacity to future market demands, as well as convertibility to accommodate new products introduced by customers. They also highlight the need for diagnosability through optimal integration of quality inspection within manufacturing systems and stress the significance of customization by designing systems around product families. Additionally, they advocate for maximizing productivity through the reconfiguration of operations and the reallocation of tasks across machines, alongside implementing effective maintenance strategies that balance machine reliability and overall system throughput. These principles subsequently serve as a framework for classifying studies, the works presented in this thesis falls under the principle 5 category.

A review that stands out for its thorough explanation and clarification of the RMS paradigm, making it particularly insightful is that of Bortolini et al., 2018. Unlike other literature reviews that often focus on a specific aspect of RMS, this study takes a broader approach, covering multiple areas and topics. Additionally, instead of rigidly categorizing or classifying studies, it introduces the concept of research "streams," emphasizing that many works overlap across these streams or subcategories. The authors identify five emerging and promising research streams, spanning from conceptual models to empirical applications, offering a comprehensive perspective on RMS research.

Our work can be categorised as belonging to Stream 4, specifically within the planning and scheduling category. This encompasses the three sub-categories: scheduling in RMS, process plan generation in RMS, and production planning for RMS. However, with potential future extensions, it can be argued that our project has the potential to be applicable across a wide range of contexts, particularly within the Stream 3 sub-categories. Additionally, Stream 5 represents an interesting avenue for further investigation, particularly given the dearth of research in this area, as highlighted by the authors.

The papers discussed in Yelles-Chaouche et al., 2021 review are optimization problems in RMS. In the beginning, the authors give a brief explanation of the fundamentals of RMS. Next, as to objective functions are important in optimization problems, they state that there are two types of Objective Functions: Machine level and systems level. The first considers machine-related functions (machine operating cost, time, operational capability, etc.,). Meanwhile, the latter deals with system-related ones (system purchase cost, time, reliability, modularity, etc.). For solution approaches, we can find two main categories, exact methods such as Mathematical modelling (ILP, MILP, etc.), DP, and constraint generation. The second consists of heuristic methods, mainly meta-heuristics (Genetic Algorithms, Simulated Annealing, Ant Colony Optimization, etc.,). Then the authors explain their classification of optimization problems, with four categories: RMS design, Production planning and scheduling, Layout design, Line balancing.

Chapter 4 can be classified as belonging to the category of RMS design, specifically within the process planning subcategory. In the case of chapters 5 and 6, however, it can be argued that the problems treated are more accurately described as ones of production planning and scheduling optimisation.

Table 3.5 summarizes the discussed literature reviews with their relative classifications.

An additional Literature review that deserves to be mentioned is Skärin et al., 2022. It considers sustainability with RMS, as this field has gained growing interest in the recent years. The authors emphasize that a company's success should be evaluated not only on financial performance but also on social and environmental impacts, aligning with the concept of sustainable manufacturing as a key component of global sustainable development. The authors categorized the reviewed papers according to the triple bottom line of sustainability: economic, environmental, and social. The paper concludes that while RMS can enhance responsiveness and cost efficiency, there is a significant gap in understanding how RMS contributes to sustainable manufacturing. The authors call for further research to establish clearer connections between RMS characteristics and sustainability outcomes. Which we cite as a perspective in chapter 7.

Table 3.2: Summary of classifications used in major RMS literature reviews.

Literature review	Classification
Andersen et al., 2015	Five categories of reconfigurable-manufacturing related papers, since the authors decided to consider the factory and segment levels as one: **Network, Factory and segment, System, Cell, Workstation**.
Gadalla and Xue, 2017	Four categories of RMT-related works: **Fundamentals on RMTs, Architecture design** composed of three sub-categories: (Semi-open architecture, Open architecture, Integral architecture,) **Configuration design and optimization** composed of two sub-categories: (Configuration design, Configuration optimization,) **System integration and control** composed of two sub-categories: (System integration, System control.)
Koren et al., 2018	Six categories of works based on RMS design principles: **Capacity planning by utilizing Principle 1**, **Functionality planning by utilizing Principle 2**, **Maintaining product quality by utilizing Principle 3**, **Formulating a product family by utilizing Principle 4**, **Process planning and line balancing by utilizing Principle 5**, **Optimizing maintenance operations by utilizing Principle 6**.
Bortolini et al., 2018	Five research streams: **Stream 1 Reconfigurability level assessment**. **Stream 2 analysis of RMS features.** Modularity. Integrability. Diagnosibility. Convertibility. Customisation. Scalability. **Stream 3 analysis of RMS performances.** Strategic perspective. Operational perspective. (Responsiveness. System complexity. Quality. Availability and reliability.) Sustainability perspective. **Stream 4 Applied research and field applications.** Inbound reconfigurable transportation systems. Layout problem. Product family formation for RMSs. Reconfigurable cellular manufacturing systems. RMS configuration selection. (Optimisation and sub-optimisation methods.) Planning & scheduling in RMSs. (Scheduling in RMSs. Production planning in RMSs. Process plan generation in RMSs.) **Stream 5 reconfigurability toward industry 4.0.**
Brahimi et al., 2019	Two classifications: **Classification 1, Problem Type**: composed of five categories: Process planning, Layout design, Reconfigurability, (composed of three sub-categories: Configuration selection/generation, Reconfigurable cellular manufacturing systems, Optimal machine selection,) Planning, Scheduling, **Classification 2, Solution approaches:** composed of four categories: Mathematical programming, DP, Meta-heuristics, Heuristics,
Yelles-Chaouche et al., 2021	Four categories of RMS optimization problems: **RMS design**, composed of four sub-categories: (Rotary machining system design, process planning, flow line configuration selection, RCMS/DCMS design.) **production planning and scheduling**, **layout design**, **line balancing and re-balancing**.

3.3.1 *PP-RMS*

The literature review Khan et al., 2022 address Process Planning for Reconfigurable Manufacturing Systems (PP-RMS) related papers. Unlike traditional analyses, this study introduces multiple classifications that consider both theoretical and practical aspects, enabling readers to position their work within the broader research landscape and identify promising directions for future exploration. The theoretical classification emphasizes research themes and RMS characteristics, while the practical classification concentrates on problem types and solution methodologies. This classification is summarized in Figure 3.1.
Furthermore, the paper's discussion and future trends sections provide valuable perspectives and propose potential avenues for further investigation, enhancing its contribution to the field.

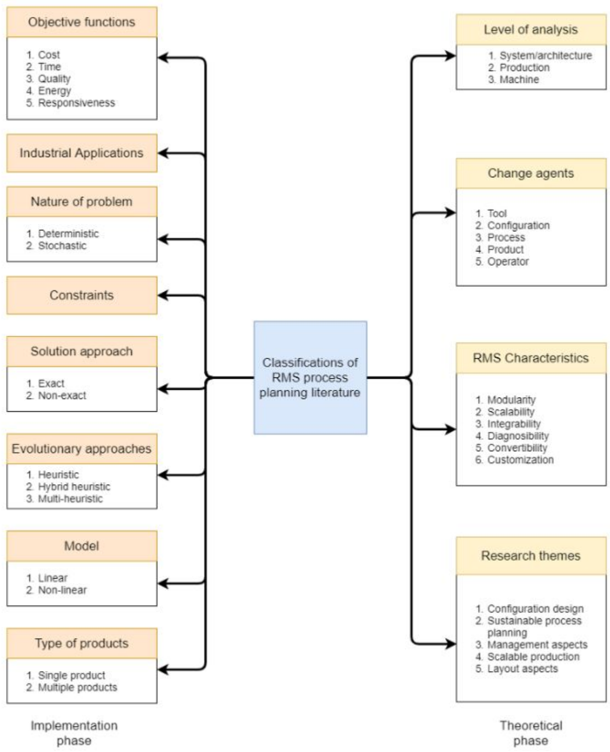


Figure 3.1: Classifications considered in the review of RMS process planning literature by Khan et al., 2022

The works of chapters 4 and 5 can be classified according to the level of analysis as production and machine level. The objective function is limited to time and cost, however, future extensions may include quality and energy. In regard to the model, we adhere to the linear model. With regard to the characteristics, we consider modularity, given that the machines and tools represent modules. We also consider the possibility of incorporating several other RMS characteristics. With regard to the solution approach, an exact methods are proposed, and non-exact methods for solving large instances. In considering evolutionary approaches, metaheuristics are also developed in chapter 5. In consideration of the nature of the problem, our approaches are limited to deterministic problems, as the complexity of stochastic problems makes them difficult to address. With regard to the product type, chapter 4 addresses single-type problems, while chapter 5 addresses a Multiple type products case. Chapter 6 is not a process planning work thus doesn't fall within this review's classification.

Regarding TAD and as mentioned in chapter 1, eight TADs are considered in the problems of chapters 4 and 5. It should be noted that we are not the first to propose this set of TADs, several studies have employed it, with the specialized TAD being designated as X in some papers, such as Y. Zhang et al., 2003, W. Li and McMahon, 2007, S. Zhang and Wong, 2016, and Azab and ElMaraghy, 2007b. However, these additional TADs were not considered in the majority of the PP-RMS papers discussed in this section and Table3.3.

3.3.1.1 *Constraints in PP-RMS*

In PP-RMS problems, multiple constraints are considered, besides machines' reconfiguration. Constraints related to operations precedence, production system capability, and machine/turret/module position have been used more often (Khan et al. 2022). Additionally, there are constraints related to demand satisfaction, investment budgets, and space and throughput.

The works discussed here primarily focus on research PP-RMS "classical works". We mean by this term works published before 2020, that consider RMTs process plan generation problem with precedence constraints between operations, and tools and TADs related constraints. While each of the mentioned works contributes to the field, we do not delve into considerable detail, we will aim to provide a concise summary. More recent works related to these studies can be found in section 3.3.1.2.

One of the earliest contributions to the field is that of Gumasta et al., 2011. The authors focus on evaluating different RMS configurations, proposing a reconfigurability index that incorporates key factors such as modularity, convertibility, scalability, and diagnosability. This index is used to assess various configurations through a Multi-Attribute Utility Theory-based approach.

In one of the earliest classical papers by Bensmaine et al., 2011, the authors explore the use of RMTs for machining a product with TADs and multiple available tools. The study unfolds in two phases: the first phase involves process planning for a single unit containing three parts, while the second phase extends the approach to multiple units. In this phase, non-dominated solutions are used to develop a sequence of process plans. It's worth noting that this approach was similarly employed by Kazemisaboor et al., 2022.
In another classical work with RMTs and TADs, Bensmaine et al., 2012 employed the Archived Multi-Objective Simulated Annealing (AMOSA) algorithm to solve process planning problem, and used the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method to rank the resulting solutions.
Chaube et al., 2012 expanded upon the original framework developed, where each operation was assigned a single TAD, and the starting and completion times of each task were calculated. This work laid the foundation for understanding the integration of TADs in process planning.
An important advancement was introduced by Bensmaine et al., 2013c, by address-

ing a problem involving a greater number of machines than in previous studies. In this work, machine selection was treated as part of the process planning problem, as only a subset of machines would be used. This paper was also significant for being the first to incorporate a different mode of execution of operations and to explore scenarios where an operation might require multiple TADs.

In another notable contribution, Bensmaine et al., 2013b shifted focus from process planning to production planning, albeit still utilizing process planning as a core component. Their approach addressed random client demand by generating multiple process plans to meet these demands. The solution involved a combination of optimization to generate process plans for each period and simulation to generate periods based on the demands received, thus integrating both planning methods in a dynamic environment.

In a non-traditional approach focused on adaptive setup planning, which is a type of process planning but incorporates more mechanical details than usual, and although it touches upon scheduling, Mohapatra et al., 2013 work does so in a rather unclear manner. The paper features three illustrative examples, which are insightful, but it should be noted that the machines in question (3-axis, 4-axis, and 5-axis machines) are not reconfigurable.

Building on their previous works Bensmaine et al., 2013a, introduce a unique combination of process planning and load balancing, where process planning accounts for the load balance of previously produced products within the RMS. It also introduces a new execution mode with a "machine zero," functioning as an entrance station for parts, marking a development in RMS design.

In another contribution, Bensmaine et al., 2014a focus on IPPS withi RMS, a concept that had not been explored extensively in previous works. Their proposed heuristic (online approach) for IPPS is compared to the traditional process planning followed by scheduling approach (also named offline approach) and demonstrates superior performance. Two key parameters are calculated in the heuristic which are machine availability and operation setup time. Despite its contributions, the work had limited extension papers, even though the authors mention several promising avenues for further research.

Bensmaine et al., 2014b present another noteworthy work in machine selection. What sets this work apart is the inclusion of machine acquisition costs, adding a financial consideration to the machine selection process that had been overlooked in previous studies. However, it's important to note that this work does not involve machine changeover, which might be a limitation when considering the full scope of machine reconfiguration in RMS.

Returning to the classical PP-RMS works with a focus on machine selection, Benderbal et al., 2015 introduced a robustness index, which quantifies the time lost in the event of machine unavailability. While the computation of this index is complex and time-consuming, it remains a valuable contribution to the model, and the authors successfully manage to solve the problem despite this challenge.

The same authors in Haddou-Benderbal et al., 2016 offered a different perspective by addressing the real-time status of the shop floor. Their work involves a set of available machines on which a pre-defined process plan is executed, though not all machines are selected. The approach includes a heuristic that generates a new

process plan each time a machine fails, and it also incorporates re-process planning when the machine is repaired. This paper builds on earlier work by the same authors and introduces the robustness index, similar to previous studies, while also incorporating real-time adjustments in process planning.
Dahane and Benyoucef, 2016 present a relatively traditional work, but with some key differences. The work focuses on process planning with machine selection, marking the first work where modular machines and maintenance considerations are incorporated. Notably, this study excludes tools from the model and introduces a machine reliability consideration along with a reconfiguration effort index, both of which could serve as useful elements for future extensions of the work.
In another contribution to the field, Haddou Benderbal et al., 2017 focuses on machine selection integrated with process planning, and the authors explanation of the flexibility index adds significant value. The flexibility index captures potential alternative routes in the event of machine failure, although not in real-time process planning.
A work that integrates process planning with layout design was presented in Haddou-Benderbal et al., 2017, which deviates from traditional machine placement considerations. Rather than focusing on the arrangement of machines, the study concentrates on selecting which machines to use. The authors present three objectives: balancing machine usage, maximizing alternative process planning, and minimizing layout evolution effort. This work leans more towards a machine selection problem, rather than a traditional layout design problem.
An extension of the work was done in Benderbal et al., 2018, but this time distinct from traditional process planning problems, by focusing on layout design. In this work, process plans are used as inputs to determine machine positions. The authors first assign an importance index to each machine, which is then used to calculate penalties for constraint violations. While this approach presents promising possibilities for further development, the authors' conclusion that the heuristic did not perform well on medium-sized instances raises concerns, leaving room for expected future improvements or extensions.
Later Haddou Benderbal et al., 2018 return to a more familiar framework, combining machine selection and process planning with modularity considerations. Their study focuses on minimizing time and cost, alongside maximizing a modularity index composed of five aspects: module commonality, availability, suitability for a product, diversification of machine usage, and fulfilment of common features by modules. Additionally, they account for module change time and cost. The paper is notable for its comprehensive literature review, citing 72 sources, and offers intriguing potential for extending the research to multi-unit problems. However, the added complexity of modularity may render a mathematical model infeasible, as the authors themselves suggest.
Touzout et al., 2018 contribute a typical process planning study, incorporating objectives related to time, cost, and the novel inclusion of Greenhouse Gas (GHG) emissions. This is the first work to propose a mathematical model with an exact solution approach for this problem. Notably, it is also the first study to test multiple-sized instances. The authors combine an Iterative Multi-Objective Integer linear programming (I-MOILP) model with AMOSA to enhance solution quality,

however, the improvements achieved were modest, signalling the need for further refinement in the approach. In a subsequent study, Touzout and Benyoucef, 2019b introduce the concept of "triplets," where machine, configuration, and tool indices are grouped together. The authors propose three approaches to solve the problem: I-MOILP, AMOSA, and NSGA-II, focusing on multi-objective optimization with three objectives: time, cost, and GHG emissions. The study concludes that NSGA-II outperforms AMOSA and highlights the importance of mutation percentage, noting that higher mutation rates lead to better solutions.

Khezri et al., 2020 focus on process planning. They propose an MILNP model that addresses the process planning problem at two levels: the lower-level generates process plans based on part requirements and energy loss objectives, while the upper-level diagnoses the reliability of selected machines and tools. Despite the slightly disorganized mathematical model, this work is valuable as a conference paper offering insight into process planning and reliability diagnosis.

Massimi et al., 2020 tackle a process planning problem within the context of RMS design, specifically focusing on the selection of machines. A distinctive feature of their work is the modular structure of RMTs, which consists of basic and auxiliary modules, with the objective of minimizing energy consumption. They present a non-linear mathematical formulation with six decision variables to model the problem and utilize a heuristic to explore all feasible solutions, starting with the definition of operation sequences. After solving four scenarios, the authors suggest that metaheuristics are necessary for solving larger instances of the problem.

Further advancing this research area, Ameer and Dahane, 2023 proposed a method that considers setup and process planning constraints, focusing on modular RMTs and fixtures. Their two-stage approach begins with a heuristic for generating setup plans and assigning fixtures for each setup. In the second stage, a GA is used to optimize the process plan, aiming to minimize total costs associated with the required operations.

Table 3.3 provides a summary of the reviewed articles a classification based mostly on considered constraints.

Table 3.3: Summary of papers from PP-RMS (Process planning in reconfigurable manufacturing systems) literature.

Paper	Objective functions			Product	Machine	Planning		Units	Solution Approach	Size of largest instance solved
	Time	Cost	Other			Process planning	Production planning			
Massimi et al., 2020			Energy	Two products of 6 and 10 operations	RMT	✔		Single	Heuristic that solve a MINLP	4 scenarios the biggest one has 10 operations and 8 machines
Khezri et al., 2020	✔	✔	Sustainability	two parts composed of three operations each	RMT	✔		Multi	MINLP model	2 parts, 3 operations each, 3 stages, 3 machines.
Touzout and Benyoucef, 2019b	✔	✔	GHG emissions	Part composed of features	RMT	✔		Single	I-MOILP & AMOSA & NSGA-II	Multiple sizes that goes from three operations and machines to 100 operations 20 machines
Touzout and Benyoucef, 2018	✔	✔	GHG emissions	Machining of a part composed of operations	RMT	✔	✔ but not really mentioned	Single	I-MOILP and AMOSA and some three other heuristics	Multiple sizes (goes from 4 operations, 4 units to 20 operations, 10 units)
Touzout et al., 2018	✔	✔	GHG emissions	Machining of a part composed of operations	RMT	✔		Single	I-MOILP and AMOSA and A hybrid heuristic combining the two	Multiple sizes (goes from 3 machines, 4 operations to 10 machines, 20 operations)
Haddou Benderbal et al., 2018	✔	✔	Modularity	Machining of a part composed of features	Modular RMT	✔	machines and modules selection	Single	AMOSA to find optimal processes and TOPSIS to rank them	
Benderbal et al., 2018			Penalty (to what extent the layout obtained violates the constraints)	Machining of a product composed of operations	RMT	it's considered as an input	Layout design	Single	Heuristic	6 machines, 9 positions, 12 operations
Haddou-Benderbal et al., 2017			Average machine usage & Impact of machine unavailability & Layout evolution effort	Three parts of the same product family composed of operations	Modular RMT	✔	layout design	Single of each part	AMOSA	10 machines, 3 parts, 14 different operation
Haddou Benderbal et al., 2017	✔		Flexibility	Machining of a product composed of features	Modular RMT	✔	machine selection	Single	NSGA-II and TOPSIS	10 machines, 22 configurations, 4 features, 15 operations, 5 tools
Dahane and Benyoucef, 2016	✔	✔	Reconfigurability	Machining of a product composed of features	Modular RMT	✔		Single	NSGA-II	6 machines, 13 configurations, 3 features, 12 operations
Haddou-Benderbal et al., 2016	✔		Robustness	Machining of a product composed of features	RMT	✔ (real time)		Single	NSGA-II	10 machines, 3 features, 12 operations, and at least 4 tools

Paper	Objective functions			Product	Machine	Planning		Units	Solution Approach	Size of largest instance solved
	Time	Cost	Other			Process planning	Production planning			
Benderbal et al., 2015	✔		Robustness	Machining of a product composed of features	RMT	✔		Single	NSGA-II	10 machines, 23 configurations, 3 features, 12 operations, and 4 tools
Bensmaine et al., 2014b	✔	✔		Machining of a part composed of subparts	RMT	✔	Embedded	Single	NSGA-II	10 machines, 3 subparts, 10 operations,
Bensmaine et al., 2014a	✔			machining one unit of two products of different type	RMT	✔	✔	Single of each product type	Heuristic using discrete event simulator and for the offline approach compared they used AMOSA and GA	3 machines, 8 configurations, 2 products, 20 operations, and 10 tools
Bensmaine et al., 2013a	✔		Machine Load	Machining of a product composed of features	RMT	✔	Load Balancing	Single but with consideration of of other units	AMOSA	3 machines, 3 configurations each, 3 features, 11 operations,
Mohapatra et al., 2013	✔	✔	Machine utilisation	3 example parts composed of features with there mechanical drawings and details	3-axis, 4 axis, and 5-axis based machine tools	✔ with adaptive setup planning	✔	Single	Artificial immune system AIS and GA	Part 1 with 14 features, part 2 with 22 features, part 3 with 31 features and 6 primary locating surfaces
Bensmaine et al., 2013b	✔	✔		Machining of a product composed of features	RMT	✔	✔	Single (repeated)	GA to generate process plans and Discrete event simulation to generate periods	3 machines, 3 configurations each, 3 features, 11 operations, and maybe 4 tools
Bensmaine et al., 2013c	✔	✔		machining of a part composed of features	RMT	✔		Single	NSGA-II	10 machines, 22 configurations, 3 features, 12 operations, and 5 tools
Chaube et al., 2012	✔	✔		machining of a product composed of three parts	RMT	✔	Embedded	Single	NSGA-II	3 machines, 9 configurations, 3 parts, 12 operations, and 7 tools
Bensmaine et al., 2012	✔	✔	Reconfiguration effort		RMT	✔	Embedded	Single	AMOSA and TOPSIS	
Bensmaine et al., 2011	✔	✔		machining of a product composed of three parts	RMT	✔	Embedded	Single and Multi	NSGA-II (twice)	3 machines, 3 configurations each, 3 parts, 11 operations, and for multi-unit 8 units

3.3.1.2 *Solution methods for PP-RMS*

In our view, process planning is not as closely aligned with operations research as scheduling or lot-sizing. It appears to us that process planning literature is less concerned with issues such as NP-hardness and integrity gaps, ... which are more prevalent in the field of operations research. To illustrate, a search on Google Scholar for the term "lot-sizing" in September 2024 returned 58,300 results, while the same search for "process planning" gave 274,000 results. A search for "lot-sizing" and "NP-hard" gives 7,800 results, whereas a search for "process planning" and "NP-hard" returns only 4,610. The discrepancy between process planning and operations research communities cannot be justified by the assumption that process planning is straightforward and simple from an operations research perspective. In fact, the simplest lot-sizing problem, SILSP of Wagner and Whitin, is still a topic of study in operations research today. we believe that this discrepancy is the primary obstacle to the advancement of process planning in comparison with other production planning-related fields.

PP-RMS is an NP-Hard problem , That's why most authors have opted for heuristic approaches for solving the problem, where it was found by Khan et al., 2022 that up to 60 % of the articles reviewed in their paper have used only heuristic solution approaches, including metaheuristics.

The GA and its multi-objective optimization variant, the NSGA-II (Deb et al. 2002), are widely employed metaheuristics for addressing PP-RMS. In a study Dou et al., proposed a GA as a solution method to minimize the capital cost associated with configuring a single-part flow line in an RMS context Dou et al. 2011. Similarly, Chaube et al., 2012 employed NSGA-II to optimize process planning, with the objective of minimizing total cost and time. These costs and times are dependent on machine and tool usage, as well as machine, configuration, and tool changeovers. Bensmaine et al., 2013c later extended the problem by incorporating machine selection, where the available number of machines exceeds the required amount. In this scenario, a process planner must determine which machines to select. However, their analysis did not consider the cost associated with selecting a machine, which could be considered a limitation.

In their study, Haddou Benderbal et al., 2018 used AMOSA to solve the PP-RMS, with non-dominated solutions ranked via the TOPSIS method. Subsequently, Khettabi et al., 2021 conducted a comparative analysis of NSGA-II with NSGA-III, a WGA, and a RWGA, with the objective of minimizing time, cost, GHG emissions, and hazardous liquid waste. The findings indicated that NSGA-III exhibited superior performance.

Heuristics, particularly those based on mathematical programming, have been relatively understudied in the context of solving PP-RMS problems. The most pertinent studies are those of Yazdani et al., 2022 and Massimi et al., 2020. In their study, Yazdani et al., 2022 compared a Lagrangian relaxation-based heuristic with a mathematical model. However, their focus was on integrated process and production planning, which differs from the problems discussed here (it will be discussed in the next section).

In addition to time and cost, the objective considered in the study of Massimi et al., 2020 was the minimization of energy consumption through the implementation of

an exhaustive search heuristic, which entails the exploration of all feasible operation sequences. However, the approach was only applicable to small instances (10 operations and 8 machines), making it impractical for larger problems. This may explain the limited use of mathematical-based heuristics for solving PP-RMS problems, as they have not demonstrated effectiveness for larger instances.
This dearth of heuristic utilization is apparent in the domain of RMS optimization in general, as observed by Yelles-Chaouche et al., 2021, no studies have employed mathematical programming heuristics for tackling these problems. The lack of interest in such heuristics can be attributed to two factors: the complexity of multi-objective optimization in RMS and the challenge of designing heuristics that are both general and specific enough to address diverse problems effectively.

A notable shortcoming of the previously mentioned studies is the absence of effective performance indicators for heuristic solutions, which has resulted in the inability to determine the optimal or Pareto front solutions for the tested instances, and thus the quality of solutions generated by heuristics can not be precisely assessed (Luo et al., 2022). As indicated by Khan et al., 2022, exact solution methods were employed in only 33% of the reviewed works, predominantly through solvers, and only 7% utilising both exact and approximate techniques, with exact approaches typically constrained to small PP-RMS problem instances (See the last column of Table 3.3).

In their study, Touzout and Benyoucef, 2019b addressed a sustainable RMS-PP problem through the application of an exact method (I-MOILP) and two metaheuristics (AMOSA and NSGA-II). The I-MOILP model, based on two main decision variables (Two main decision variables model (2MDV)), was evaluated on small instances (The model is presented with some changes in chapter 4 section **??**), while the metaheuristics were applied to instances of varying complexity.
Subsequently, Khezri et al., 2021, among other works, used the same model. They employed it with the augmented e-constraint method, comparing it with SPEA-II and NSGA-II. The evolutionary approaches were observed to perform better and taking less computational time, however, the tests were limited to a single instance due to the high computational demands of RMS-PP. Despite the fact that several studies have employed the formulation proposed by Touzout and Benyoucef, no comparison has been made with other models.

From this literature review and Table 3.3, we have identified four research gaps:

- There are no lower bounds for PP-RMS problem in the literature.
- There is a lack of use of mathematical-based heuristics.
- A comparison between an exact and an approximate solution method is a crucial yet often overlooked aspect of solution methods evaluation.
- A comparison of the mathematical formulation from the literature with alternative formulations was not undertaken, despite the possibility of further improvements.

The studies discussed in this subsection focus mostly on single-product process planning, aligning with the dominant trend in PP-RMS research. According to Khan et al., 2022, approximately 70% of the reviewed articles address this type of problem. However, multi-product process planning provides a more realistic representation of manufacturing environments and holds greater potential for improving the efficiency of RMS.

3.3.2 *Multi-Product Process Planning*

With the growing demand for diversified products, there is more focus on optimizing multi-product production. Therefore, the category of "*integration of process planning with production systems*" from Xu et al., 2011 classification (see section 3.2), has gained increasing attention.

Researchers have explored various problems involving the integration of production and process planning, including those closely related to Multi-Product process planning (MPPP) problem discussed in Chapter 5. Similar problems arise in studies focused on multi-product assembly lines. For instance, Kant et al., 2020 addressed the formation of reconfigurable machine cells and the sequencing of products on an assembly line, with the objective of minimizing the total time and effort required for reconfiguration.
In another study, Stief et al., 2023 investigated the simultaneous optimization of resource selection, operation and machine allocation, and product sequencing. They proposed a MILP model to address this problem and employed the normalized weighted sum method for efficient solution generation.

Table 3.4 summarizes key research contributions on MPPP within RMS, highlighting their classifications and distinguishing features, as well as our two works presented in chapters 4 and 5. It details the types of decisions integrated with process planning, such as product sequencing, and specifies whether the products considered are identical or of varying types. The scope of reconfiguration is also analysed, differentiating between system-wide and machine-specific adjustments, with our study focusing on machine-level modifications for RMTs. Additionally, the table categorizes the studies based on optimization objectives, solution methodologies, and the types of approaches used (exact or approximate).

IPPS for RMS has been extensively studied, with researchers proposing innovative methods to address the complexity of this problem, which combines two NP-hard subproblems. For instance, Gao et al., 2021 proposed an online approach that integrates process planning, scheduling, and layout design decisions into a unified framework. Similarly, Bensmaine et al., 2014a focused on minimizing makespan in the IPPS problem, comparing the effectiveness of online and offline strategies. Their results showed that the online heuristic consistently outperformed the offline method in solution quality. These findings have inspired the prioritization of online approaches in this research to tackle the challenges of the MPPP problem.
In addition to integrating scheduling, numerous studies have explored the combination of process planning and product sequencing. This domain is referred to as Multi-Unit process planning (MUPP) or Multi-Part Process Planning, depending on whether multiple product types are considered. For example, Touzout and Benyoucef, 2019a addressed a multi-unit, single-type process planning problem by utilizing the NSGA-II metaheuristic within an offline framework. Their approach involved independently conducting process planning for each product, followed

Table 3.4: Classification of MPPP (Multi-Product process planning) related papers with two chapters 4 and 5 and a classification based on both theoretical and practical factors.

Article		Stief et al., 2023	Yazdani et al., 2022	Gao et al., 2021	Bensmaine et al., 2014a	Bensmaine et al., 2011	Touzout and Benyoucef, 2019a	Kazemisaboor et al., 2022	Musharavati and Hamouda, 2012	Chapter 5	Chapter 4
Process planning combined with	Sequencing	✔				✔	✔	✔	✔	✔	
	Scheduling			✔	✔						
	Production planning		✔	✔							
	System Design	✔		✔							
Product type	Single					✔	✔	✔		✔	✔
	Multiple	✔	✔	✔	✔				✔	✔	
Reconfiguration level	System	✔	✔	✔					✔		
	Machine			✔	✔	✔	✔	✔		✔	✔
Optimization	Mono				✔				✔		✔
	Multi	✔	✔	✔		✔	✔	✔		✔	
Approach	Online	✔	✔	✔	✔				✔	✔	
	Offline				✔	✔	✔	✔		✔	
Solution Method	Exact	✔	✔	✔						✔	✔
	Approximate		✔	✔	✔	✔	✔	✔	✔	✔	✔

by sequencing the resulting plans. As highlighted in Chapter 1, online approaches aim to concurrently integrate these decisions, thereby offering potential for enhanced efficiency and streamlined operations.

In addition, researchers such as Kazemisaboor et al., 2022 and Bensmaine et al., 2011 have investigated MUPP using an offline approach. Their methodology involved a two-phase decomposition comprising optimisation and simulation. In their study, Kazemisaboor et al., 2022 found that NSGA-II was the most effective metaheuristic among those tested, including AMOSA and Multi-Objective Particle Swarm Optimization. This finding reinforces the popularity of NSGA-II in PP-RMS (Khan et al., 2022). However, MOEA/D has recently emerged as a promising alternative. In contrast to NSGA-II, MOEA/D avoids the calculation of crowding distances, thereby offering computational advantages (Q. Zhang and Li, 2007). The work in chapter 5 builds on these findings by comparing these two metaheuristics for MPPP in an RMS environment.

Musharavati and Hamouda, 2012 investigated the multi-part flow line planning problem using simulated annealing, focusing on a bi-objective optimization approach to minimize costs while maximizing throughput. Their model featured a flow line with serial stages supported by parallel reconfigurable machines, though the use of unidirectional material handling equipment restricted the system's flexibility. To overcome this limitation, X. Gu and Koren, 2024 suggested incorporating a return conveyor or gantry into traditional flow lines. This modification significantly improved the flexibility of the system, aligning it more closely with the versatile nature of true RMS configurations.

While substantial advancements have been made in MUPP, many studies focus on identical products, overlooking the variation within a product family. This emphasis underscores the *scalability* characteristic of RMS but often neglects *convertibility*the capability to adapt to different product types within the same family. Bridging this gap requires approaches that address both scalability and convertibility, enabling RMS to effectively handle diverse and evolving production demands. From the presented literature review and table 3.4, we have identified four research gaps:

- Research into MPPP as an extension of the MUPP problem remains unexplored.
- The MOEA/D metaheuristic has not been applied to PP-RMS problems, nor has it been directly compared to NSGA-II within this domain.
- While effective exact solution methods for small-scale MUPP problems are absent in the reviewed literature, there is also no precedent for employing online solution approaches in this context.
- The utilisation of an online solution approach for PP-RMS in a multi-product context, with consideration of tools and TADs-related constraints, is a novel approach that has yet to be explored.
- The application of online methods to address PP-RMS for multiple products, considering tool and TAD constraints, is an innovative direction that has not been pursued.

3.3.3 *Lot-sizing with RMS*

This section presents the most relevant literature on lot-sizing problems with reconfigurable machines. In addition to including literature that explicitly mentions the term "RMS", other relevant paper that consider the *changeability* of machines is also considered.

One of the earliest works to consider RMS with batch sizing is that of Abbasi and Houshmand, 2011, who developed a mathematical model for optimising lot-sizing in RMS using a genetic algorithm. The model focuses on efficiently adjusting production rates and configurations to respond to fluctuating market demands while minimising costs associated with changeovers and inventory holding. However, the problem considered was not discrete but rather continuous and stochastic.

Furthermore, the authors adjust production rates rather than quantities. For each product family, they define a set of configurations for processing.
Table 3.5 summarizes the most pertinent articles along with a classification and a comparison with the work proposed in this chapter.

Studying the integrated lot-sizing and job-shop scheduling problem with reconfigurable machines, Rohaninejad et al., 2022 developed an MILP for solving the problem with the objective of minimizing the total cost including Inventory and setup costs (time varying), and production, configuration changeover costs (non time varying). Later, the same authors proposed an MILP and a decomposition heuristic in Rohaninejad et al., 2024 for solving a similar problem, however this time the scheduling considered was "*semi-flexible*[2]" since even though operations are eligible to be processed on different configurations, there's only one configuration capable of processing an operation. Another limitation in this work is that there's no reconfiguration cost incurred when a reconfiguration occurs.

The problem of lot-streaming is similar to that of lot-sizing, however only one period is considered and the demands must be subdivided across multiple lots on different machines. The lot-streaming problem with reconfigurable machines was addressed by Fan et al., 2023 and Fan et al., 2024 through the development of a MILP model coupled with a matheuristic and a variable neighbourhood search algorithm. Although these studies successfully address the lot-splitting of demands across multiple lots, they fail to address the other decisions inherent to lot-sizing problems, namely to balance production and storage.

From the literature review and table provided compared to previous sections, it can be seen that the field of lot-sizing problems is relatively less studied in the context of reconfigurability compared to other areas of production planning. Furthermore, no studies have been conducted on the SILSP problem with reconfigurable machines, with total cost considered as the objective function and with startup costs and MOQ.

2 We name it semi-flexible since it is not a simple job-shop nor a flexible one.

Table 3.5: Summary of the relevant lot-sizing literature that considers RMS or multiple modes of jobs processing.

Papers	**Explicitly mentions**	**Item**	**Periods**	**Costs and times considered**					**Objective function**	**Model**	**Bucket**	MOQ	**Reconfigura -tion level**	**Workshop**	**Solution method**
				Pro	Inv	Set	Rec	Other							
Rohaninejad et al., 2022	RMS LS	Multi	Multi	C/T	C	C	C/T		Sum of costs	Deter.	Big		Machine	Flexible Job-shop	MILP
Fan et al., 2023	RMS LS	Multi	Single	T			T		Total weighted tardiness	Deter.	Small		Machine	Flexible Job-shop	MILP & Matheuristic & Metaheuristics
Roshani et al., 2023	LS	Multi	Multi	C/T	C	C/T			Sum of costs	Deter.	Big		Machine	Single Machine	MILP & Fix-relax & Metaheuristic
Rohaninejad et al., 2024	RMS LS	Multi	Multi	C/T	C	C	T		Sum of costs	Deter.	Small		Machine	Semi-Flexible Job-shop	MILP & Decomposition Hr
Furlan et al., 2024	LS	Multi	Multi	T	C	C/T			Sum of costs	Deter.	Small		Machine	Flexible Flow shop	MILP & Heuristics & Metaheuristic
Fan et al., 2024	RMS LS	Multi	Single	T			T		Total weighted tardiness	Stoch.	Small		Machine	Flexible Job-shop	MILP & Matheuristic & Metaheuristics
Jiang et al., 2024	RMS LS	Multi	Multi	T	C		C/T		Makespan	Stoch.	Big		Machine	Flow shop	MILP & column-and-constraint generation Algo.
Chapter 6	RMS LS	Single	Multi	C/T	C		C/T	Str	Sum of Costs	Deter.	Big	✔	Machine	Single Machine	MILP & DP

C: Cost, T: Time, LS: Lot-sizing,

3.4 CONCLUSION

This chapter presented an overview of the literature on production planning and RMS. The three main research streams that were emphasised were PP-RMS, MPPP with RMS, and lot-sizing with RMS.

Although the problem of lot-sizing has a substantial body of literature, there are only a limited number of applications that consider reconfigurable machines. Notably, even in the context of SILSP, there is a dearth of literature that considers startup costs in conjunction with MOQ.

Additionally, regarding the literature on the integration of process and production planning decisions, particularly with regard to product sequencing, there is a dearth of studies that have considered the sequencing of products of different types, especially in a multi-objective context. Moreover, a comparison of the exact and approximate solution approaches is crucial in this field. As for approximate methods, the most frequently employed approach in PP-RMS is NSGA-II. However, there are other potential contenders that warrant further investigation, such as MOEA/D, which is notably absent from the existing literature.

Furthermore, an examination of the literature on Single-Unit Process Planning (SUPP) reveals that heuristics are rarely employed as a method, despite their suitability for process planning problems. Additionally, an analysis of the largest instances typically solved indicates a dearth of rapid methods that could facilitate a deeper understanding of larger problems, such as LBs. These literature gaps will be tackled in the following chapters.

4 SINGLE-UNIT PROCESS PLANNING WITH RMS

This chapter presents the single-product single-unit process planning problem, which was studied within a workshop composed of RMT. In order to address this problem, a number of approaches are presented. The proposed methodology comprises two O-1 LP mathematical models, two Lower Bound (LB), and two mathematical-based heuristics that can be used as Upper Bound (UB). To evaluate the quality of the proposed approaches, we conducted extensive tests and provided a comprehensive discussion of the results, offering insights that emerged from this analysis.

The remainder of this chapter is organised as follows: Section 4.2 presents the problem under consideration, followed by a description of the mathematical models in Section 4.3. Next, section 4.4 presents the lower bounds, while section 4.5 outlines the heuristic procedure with its two variants (Heuristic 1.0 (H1.0) and Heuristic 1.1 (H1.1)). Section 4.6 provides numerical examples and a detailed analysis of the obtained results. Finally, section 4.7 offers concluding remarks and outlines future work directions.

4.1 INTRODUCTION

In a reconfigurable environment, process planning serves as a crucial "*soft-type enabler*" for changeable manufacturing systems Azab and ElMaraghy, 2007b. In fact, effective process planning is essential for ensuring the smooth processing of operations in any manufacturing system Azab and ElMaraghy, 2007a. The importance of process planning is also evident at both small and large scales.

As previously stated in chapter 1 and 3, there are two principal categories of process planning: retrieval and generative. While generative process planning is capable of providing optimal process plans, practitioners often opt for retrieval planning due to the considerable computational time required for the generative approach. Computational power has advanced considerably in recent years, yet process planning remains a time-consuming task, even without consideration of complex manufacturing setups. As a result, there is a clear need for effective solution methods for generative process planning.

In our work, the focus will be on solving the process plan generation problem, which can be considered as a subproblem in the generative approach (Sadaiah et al. 2002).

The varying problem sizes and levels of technical details incorporated into the process plan, which depends on the type of manufacturing environment considered, introduce an additional layer of difficulty Rembold et al., 1993. As a result, some problems become challenging to solve with exact solution methods. For instance, in a mass production setting where operators are typically not highly skilled and the factory layout is product-oriented, the process plan needs to be comprehensive and include specific details about each individual operation Scallan, 2003. In such a context, the number of possible combinations and their permutations grows exponentially, making it more difficult to generate even a feasible solution, let alone the optimal one ElMaraghy, 1993.

The single-product single-unit process planning problem has been shown to be NP-hard. The well-known NP-hard Travelling Salesman Problem (TSP) can be reduced to an instance of our problem, where for each operation, an optimal machine, configuration, and tool must be selected, with consideration of precedence constraints. These operations, thus, represent the cities to be visited, and an optimal route has to be selected, with the changeover times representing the distance between the cities (Khan et al. 2022; Khettabi et al. 2022; Khezri et al. 2021; Massimi et al. 2020).

An additional challenge in process planning is the potential infeasibility of certain part designs due to designers' limited understanding of production constraints, particularly in complex systems like RMS. It is not always easy for manufacturing teams to provide timely feedback on production time, cost, and quality of a product design, especially in dispersed environments (Toussaint and Cheng 2002). This highlights the necessity for rapid process planning solutions, where the representation of process planning knowledge also impacts the model's performance (Sormaz and Khoshnevis 1997).

The preceding sentences underscore the necessity for a combination of tools, techniques, and algorithms that guarantee the generation of effective process plans, readily available to designers, to facilitate decision-making in a reconfigurable environment, and the generation of alternative process plans when needed, enabling further enhancement of product design ElMaraghy, 1993. This is what we try to address in this chapter.

To tackle the previously mentioned literature gaps in chapter 3 section 3.3.1.2, and as it can be seen on figure 4.1. Our work consists of solving the process plan generation problem with RMS using a mathematical model based on one main decision variable (One main decision variable model (1MDV)). We compared it to a model based on a different modelling approach, taken from the literature, based on two main decision variables (1MDV). We also developed two LB based on relaxing the problem First Lower Bound (LB1) and Second Lower Bound (LB2). 1MDV and the LB1 were used to develop a heuristic with two variants H1.0 and H1.1 to generate near-optimal (or optimal) solutions.

4.2 SINGLE-UNIT PROCESS PLANNING

In this section, we describe the process planning problem studied with its corresponding specifications. In SUPP problem addressed, a process plan is needed to

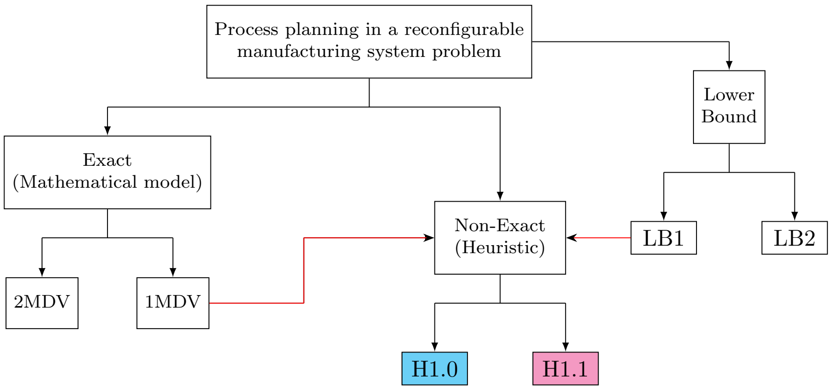


Figure 4.1: Summary of the methods proposed in this Chapter.

manufacture a part on a set of RMTs, which dictates the sequence of execution of operations and machines to be used. The process planning problem and its corresponding constraints are described in this section.
As summarized in Figure 4.4, the manufacturing of a part on a set of RMTs in the SUPP problem, requires the generation of a process plan that specifies the order in which activities must be completed and the machines that must be utilized.

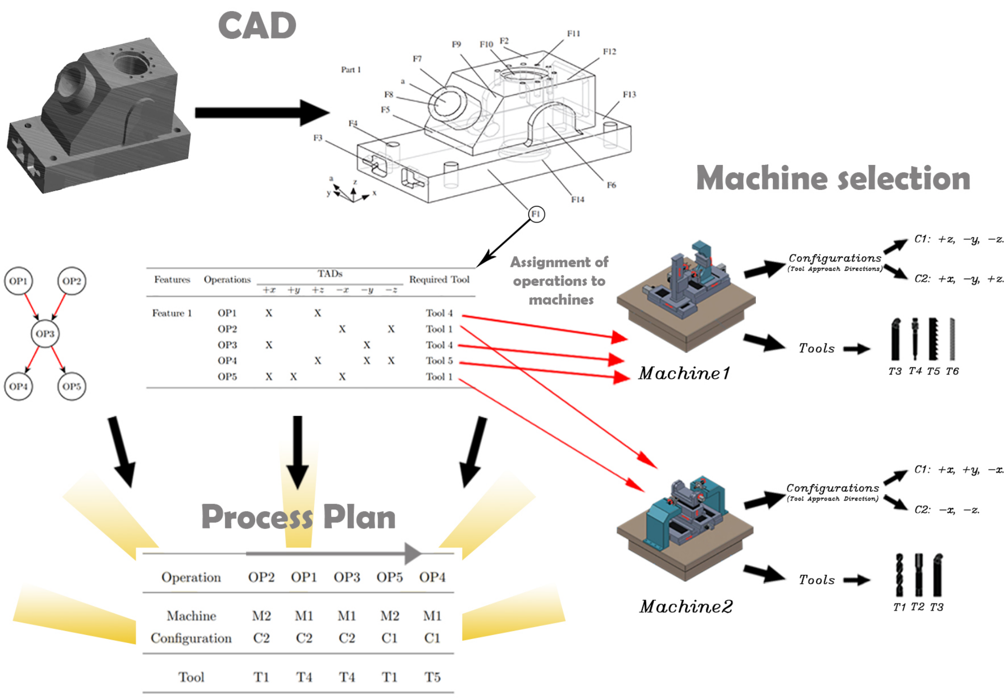


Figure 4.2: Description of process planning problem considered with the main elements and decisions taken.

The SUPP problem depicted in Figure 4.4 can be decomposed into three main steps: i) Machine selection, ii) Assignment of operations, and iii) Sequencing of operations.

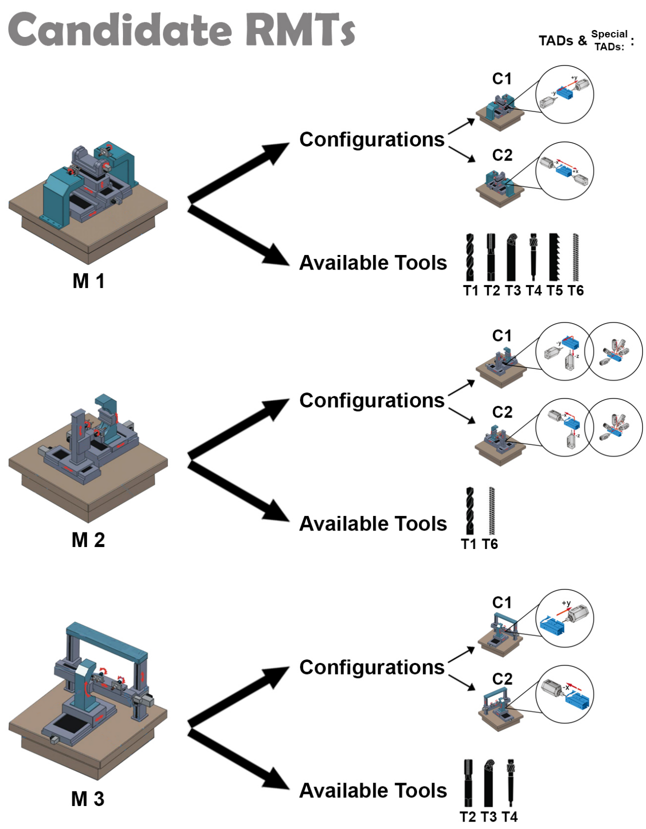


Figure 4.3: Machine selection subproblem Mechaacha et al., 2024.

4.2.1 *Machine Selection*

The operations that comprise the parts must be processed on RMTs. A precedence graph is a graphical representation that links the operations, while TADs and candidate tools are required for each operation. Conversely, there is a set of n candidate RMTs, which vary in processing speed and functionality for handling the operations. The objective is to select the most suitable RMT in order to reduce the overall manufacturing time of the part. Given that each RMT has a unique configuration, a set of TADs is available, and each RMT can only use tools from its tool magazine, it is necessary to take the operations' requirements and processing times into account in order to select the most suitable machines from the available RMTs.

Figure 4.3 illustrates a scenario comprising three RMTs, each with two distinct configurations and available TADs. It should be noted that only M2 has the unique TAD capability. Furthermore, M1 has the capacity to utilise all six tools, whereas M2 is only equipped with T1 and T6, and M3 has access to T2, T3, and T4.

4.2.2 *Assignment of Operations*

The assignment of operations to machines is a fundamental aspect of process planning. Figure 4.4 illustrates the assignment of operations, in which M1 and M3 were selected for the processing of the part with a total of five operations. In the initial stage, the machines that are capable of processing each operation are identified based on the requirements of the operations themselves. The candidate machines are represented as squares in Figure 4.4 for illustrative purposes. For instance, given that there is only one candidate machine (M3 for OP2 and M1 for OP5), no decision needs to be made for the two operations. To reduce processing times, the remaining processes were assigned to machines. The red arrows represent the final assignment.

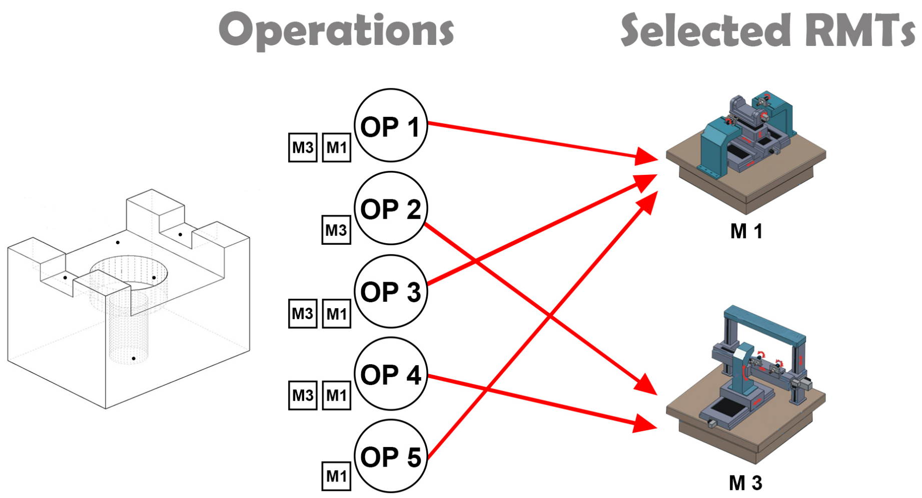


Figure 4.4: Assignment of operations to selected RMTs Mechaacha et al., 2024.

4.2.3 *Sequencing of operations*

The objective is to identify the optimal sequence of operations, with the aim of minimizing machine, configuration, and tool changeovers, while ensuring the precedence requirements of the processes are adhered to. A process plan representation is presented in Table 4.5, which may be read from left to right, column by column. The initial column indicates that OP1, executed on machine M1 configuration C2 with tool T1, is the first operation to be processed. Subsequently, M2 processes OP4 using tool T1, configuration C2, and so forth. The changeovers between machines, configurations, and tools are indicated by the arrows on the table.

There are three types of changeovers considered:

- Machine changeover: refers to the tasks involved in transferring a workpiece from one machine to another, including preparation, cleaning, transportation, and workpiece installation and removal from machines.

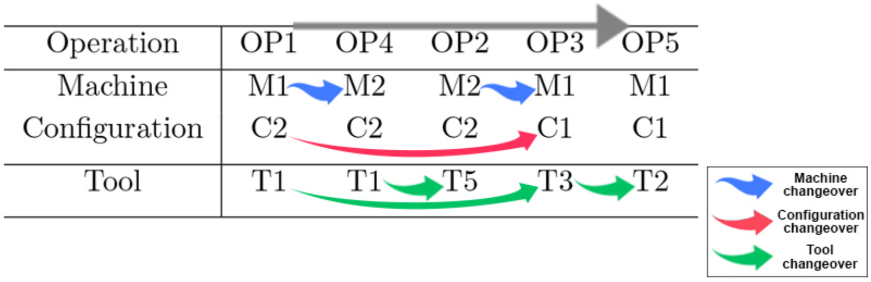


Figure 4.5: A process plan representation Mechaacha et al., 2024.

- Configuration changeover: is the process of transforming a machine's state from one configuration to another by modifying the components of the RMT or by adding modules or removing them.
- Tool changeover: refers to the operations involved in taking out the previous tool and setting up the new one.

4.2.4 *Illustrative Example*

In this example, we generate a process plan for a piece composed of three operations executed on RMTs 1 and 2 with two and three configurations respectively, we will not consider tools in this example. The data is in Tables 4.1, 4.2 ,and 4.3, and the process plan obtained is presented in Table 4.4.

Table 4.1: Processing time/possible machines

Operation	M1	M2
OP1	2	2
OP2	4	
OP3		4

Table 4.2: Machine-changing time (*tu*)

	M1	M2
M1	0	2
M2	3	0

Table 4.3: Configuration changing time (*tu*)

M1		C1	C2	
	C1	0	1	
	C2	3	0	
M2		C1	C2	C3
	C1	0	4	4
	C2	4	0	4
	C3	4	4	0

Table 4.4: The Process Plan obtained for the illustrative example.

Operation	OP1	OP2	OP3
Machine	M1	M1	M2
Configuration	C1	C2	C2

Figure 4.6 represents the Gantt chart of the process plan shown in the table 4.4. The process plan can be interpreted from left to right column by column. So first, our workpiece will be positioned in machine M1 set on configuration C1 to execute operation OP1. Next, we will execute OP2 on the same machine (M1), but this time it will be in configuration C2; this means we must change M1's configuration from C1 to C2. The final operation (OP3) will necessitate a machine change where the workpiece will be transported from M1 to M2.

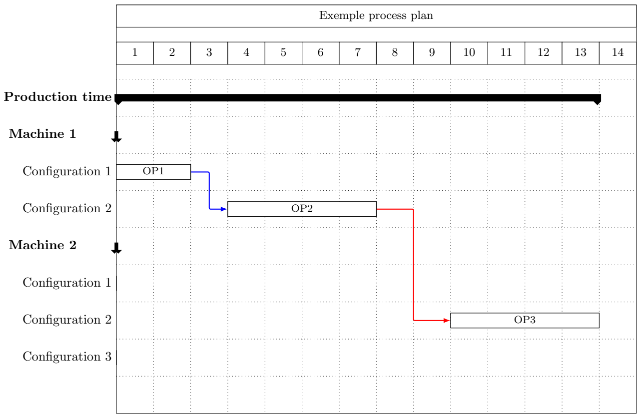


Figure 4.6: Gantt chart of the process plan obtained for the illustrative example presented.

It is important to note that the Gantt chart representation, as illustrated in Figure 4.6, is not a common method for depicting process plans in the academic process planning literature. Gantt charts are not an appropriate representation of process plans, as they are designed to illustrate the *moments* in time when operations are executed and their respective start and completion dates. The utilization of a Gantt chart in this context is justified by my belief that it offers a clearer visual representation of the sequence of operations and changeovers. It should be noted that Gantt charts are more employed in the context of scheduling, rather than in the domain of process planning, where process plans are usually presented in tables or routing sheets.

4.3 PROBLEM FORMULATION (MATHEMATICAL MODELS)

This section presents two mathematical models for solving the SUPP. The first model features a single main decision variable (1MDV), and an extension for overlapping changeover tasks is also presented. The second model includes two main decision variables (2MDV). Before proceeding to the models, the common notations and parameters utilized in both are first presented.

To formulate the problem, the following notations are used:

Indexes	
j	Index of machines
l	Index of configurations
u	Index of operations
q	Index of tools
t	Index of TADs
s	Index of positions

Sets	
J	Set of all machines
L	Set of all configurations
U	Set of all operations
Q	Set of all tools
TAD	Set of all TADs
S	Set of all positions

Parameters	
m	Number of available machines
T	Number of available tools
NC_j	Number of available configurations for machine j
NOP	Total number of operations
MT_{jq}	Equals to 1 if tool q is available on machine j
$CTAD_{jlt}$	Equals to 1 if TAD t is offered by machine j in configuration l
$OPTAD_{ut}$	Equals to 1 if TAD t is required for the operation u
OPT_{uq}	Equals to 1 if tool q is required to execute the operation u
$OPP_{uu'}$	equals 1 if operation u' has to be processed after operation u
$CCTime_{jll'}$	Configuration change time for machine j from configuration l to l'
$TCTime_{qq'}$	Tool change time from tool q to q'
$MCTime_{jj'}$	Machine change time from machine j to j'
$PrTime_{uj}$	Processing time for operation u on machine j
M	A big number

4.3.1 *First Mathematical model (1MDV)*

The 1MDV 0-1 linear programming (0-1LP) model is based on one main decision variable with five dimensions P^u_{sjlq}.

A *main decision variable* (MDV) is a decision variable from which all other decision variables may be derived. The remaining decision variables are therefore auxiliary, used to preserve the linearity of the model or facilitate its interpretation. In the following 1MDV model, MDV is P^u_{sjlq}, while the remaining variables are auxiliary. These are $Z_{jss'}$, $W_{sjj'}$, $X_{sjll'}$, and $Y_{sjqq'}$. This is because we can determine all changeovers that occur once we know the order in which each operation is performed and the machine, configuration, and tool used.

Decision variables	
P^u_{sjlq}	equals 1 if operation u is in position s executed by machine j in configuration l using tool q, 0 otherwise
$Z_{jss'}$	equals 1 if machine j is used in positions s and s'
$W_{sjj'}$	equals 1 if there is a changeover from machine j to j' between positions s and $s+1$
$X_{sjll'}$	equals 1 if there is a changeover for machine j from configuration l to l' in position s
$Y_{sjqq'}$	equals 1 if there is a changeover for machine j from tool q to q' in position s

Objective function

$$\text{Minimize: } f_{time} = PT + MCT + CCT + TCT \tag{4.1}$$

$$\text{Processing time:} \qquad PT = \sum_s \sum_u \sum_j \sum_l \sum_q P^u_{sjlq} \times PrTime_{uj} \tag{4.2}$$

$$\text{Machine changeover time:} \qquad MCT = \sum_s \sum_j \sum_{j'} W_{sjj'} \times MCTime_{jj'} \tag{4.3}$$

$$\text{Configuration changeover time:} \qquad CCT = \sum_s \sum_j \sum_l \sum_{l'} X_{sjll'} \times CCTime_{jll'} \tag{4.4}$$

$$\text{Tool changeover time:} \qquad TCT = \sum_s \sum_j \sum_q \sum_{q'} Y_{sjqq'} \times TCTime_{qq'} \tag{4.5}$$

Constraints

$$Z_{jss'} \geq (1 - \sum_{s''=s+1} \sum_u \sum_l \sum_q P^u_{s''jlq}) + (\sum_u \sum_l \sum_q P^u_{sjlq} + P^u_{s'jlq}) - 2$$
$$\forall j{=}1,..,m, \quad \forall s{=}1,..,NOP{-}1, \quad \forall s'{=}s{+}1,..,NOP \tag{4.6}$$

$$W_{sjj'} \geq \sum_u \sum_l \sum_q P^u_{sjlq} + P^u_{s+1j'lq} - 1 \qquad \forall s{=}1,..,NOP{-}1, \quad \forall j,j'{=}1,...,m \tag{4.7}$$

$$X_{sjll'} \geq Z_{jss'} + (\sum_u \sum_q P^u_{sjlq} + P^u_{s'jl'q}) - 2 \qquad \forall s,s'{=}1,..,NOP, \quad \forall j{=}1,...,m, \quad \forall l,l'{=}1,...,NC_j \tag{4.8}$$

$$Y_{sjqq'} \geq Z_{jss'} + (\sum_u \sum_l P^u_{sjlq} + P^u_{s'jlq'}) - 2 \qquad \forall s,s'{=}1,..,NOP, \quad \forall j{=}1,...,m, \quad \forall q,q'{=}1,..,T \tag{4.9}$$

$$OPP_{uu'} + \sum_j \sum_l \sum_q P^u_{sjlq} + P^{u'}_{s'jlq} \leq 2 \qquad \forall s{=}2,..,NOP, \quad \forall s'{=}1,...,s \quad \forall u,u'{=}1,...,NOP \tag{4.10}$$

$$P^u_{sjlq} \times OPTAD_{ut} \leq CTAD_{jlt} \qquad \forall s{\in}S, \forall u{\in}U, \forall l{\in}L, \forall q{\in}Q, \forall j{\in}J, \forall t{\in}TAD \tag{4.11}$$

$$P^u_{sjlq} \leq MT_{jq} \times OPT_{uq} \qquad \forall s{\in}S, \forall u{\in}U, \forall l{\in}L, \forall q{\in}Q, \forall j{\in}J, \forall u{\in}U \tag{4.12}$$

$$\sum_s \sum_j \sum_l \sum_q P^u_{sjlq} = 1 \qquad \forall u{\in}U, \tag{4.13}$$

$$\sum_u \sum_j \sum_l \sum_q P^u_{sjlq} = 1 \qquad \forall s{\in}S, \tag{4.14}$$

$$P^u_{sjlq}, Z_{jss'}, W_{sjj'}, X_{sjll'}, Y_{sjqq'} \in \{0,1\} \tag{4.15}$$

The objective function in our models is to minimize the total production time of the part, which is composed of machining time, tool changeover time, and machine and configuration changeover times, as shown in equation (4.1), and the equations from (4.2) to (4.5). Constraints (4.6) establish a link between the utilization of each machine and its subsequent position where it has been used, using the variable Z. When there is a machine changeover between positions s and $s+1$, constraints (4.7) specify that. The occurrence of configuration and tool changes is also specified by constraints (4.8) and (4.9) respectively. An operation is only executed if all its predecessors have been processed according to the constraints (4.10). Each machine configuration must have the TADs necessary to process the operations assigned to it, thanks to constraints (4.11). The constraints defined in (4.12) ensure that each machine has at least one of the necessary tools available in its tool magazine to complete the assigned operation. Each operation is performed exactly once, following the constraints defined in (4.13). The constraints defined in (4.14) indicate that a single operation is carried out at a time. The integrality constraints are represented by (4.15).

4.3.1.1 *Overlapping changeover tasks*

To illustrate the extensibility of the 1MDV model and the distinction between the process planning and scheduling activities, we will consider the following scenario. A process planner has observed that in scheduling the changeover tasks are performed by two different operators. One operator is responsible for transferring the part from one machine to the next, while the other is tasked with preparing the next machine, which involves configuration and tool changeover tasks. In this case, a potential solution to integrate scheduling considerations into the process planning model is to consider only the maximum between the machine changeover time and the configuration plus the tool changeover times, instead of the sum of the three changeovers. This problem has been previously addressed in the literature Chaube et al., 2012 (See chapter 3), however, no linear programming mathematical model was proposed to address it. In this scenario, the machine changeover and configuration/tool changeover tasks are performed simultaneously.

To the previously defined 1MDV model, we add the following decision variable:

Decision variables	
$CHTime_s$	The time of change tasks done in position s

We replace the changeover times by the total changeover time in the objective function. We replace equations (4.1) by (4.16), and equations (4.3), (4.4), and (4.5) by (4.17).

$$f_{time} = PT + TotC \tag{4.16}$$

Total change time ($TotC$):

$$TotC = \sum_{s} CHTime_s \tag{4.17}$$

And we add the following constraints:

$$\sum_{j}\sum_{j'} W_{sjj'} \times MCTime_{jj'} \leq CHTime_s \qquad \forall s{=}1,..,NOP{-}1, \tag{4.18}$$

$$\sum_{j}\Big(\sum_{l}\sum_{l'} X_{sjll'} \times CCTime_{jll'}\Big) + \Big(\sum_{q}\sum_{q'} Y_{sjqq'} \times TCTime_{qq'}\Big) \leq CHTime_s \qquad \forall s{=}1,..,NOP{-}1, \tag{4.19}$$

Theoretically, the model was correct, however, when implemented in the solver, the results were partially inconsistent. It has been observed that certain machine, configuration, and tool changeovers were unnecessarily added. This was due to the decision variable $Z_{jss'}$ being equal to one in multiple cases where it was supposed to be equal to zero (machine j was not used in positions s or s'), which had a domino effect on the other decision variables.

In the previous model, all changeover tasks were computed within the objective function, thus the problem didn't arise then. Consequently, when presented with the option, the solver would prefer to set $Z_{jss'}$ to zero rather than one. To ensure that the variable $Z_{jss'}$ is fixed to zero when necessary, the following constraints have been added:

$$Z_{jss'} \leq \sum_{u}\sum_{l}\sum_{q} P^{u}_{sjlq} \qquad \forall j \in J, \quad s=1,..,TNOP-1, \quad s'=s+1,..,TNOP, \tag{4.20}$$

$$Z_{jss'} \leq \sum_{u}\sum_{l}\sum_{q} P^{u}_{s'jlq} \qquad \forall j \in J, \quad s=1,..,TNOP-1, \quad s'=s+1,..,TNOP, \tag{4.21}$$

4.3.2 *Second Mathematical model (2MDV)*

The modeling approach developed by Touzout and Benyoucef was used to construct the 2MDV model, which is based on the two main decision variables, namely, x^k_{ij} and y^m_{jk}. If the k-th triplet is used to process the i-th operation at the j-th position, the variable x^k_{ij} takes on a value of 1. Similarly, if the m-th machine processes an operation in the j-th position using the k-th triplet, the variable y^m_{jk} takes on a value of 1. In this context, triplets represent feasible sets of tools, machines, and configurations.

Generation of triplets requires processing of data to identify every possible triplet for each machine and operation. In contrast, our approach involved direct utilisation of raw data. Consequently, modifications were necessary to the original model Touzout and Benyoucef, 2019b to facilitate the breakdown of triplets into three distinct indices: machines, configurations, and tools.

Decision variables	
x^{jlq}_{us}	1 if the uth operation is processed at the sth position using the jth machine in the lth configuration using the qth tool, 0 otherwise
y^{jlq}_{s}	1 if the jth machine is set in the lth configuration using the qth tool at the sth position, 0 otherwise
$cm_{sjj'}$	1 if between position $s-1$ and s, there has been a change between machines j and j', 0 otherwise
$cc^{lql'q'}_{sj}$	1 if between position $s-1$ and s, there has been a change from configuration l or tool q to l' or q' of machine j, 0 otherwise

Objective function

$$\text{Minimize: } f_{time} = PT + MCT + CCT + TCT \tag{4.22}$$

$$\text{Processing time:} \quad PT = \sum_s \sum_u \sum_j \sum_l \sum_q x_{us}^{jlq} \times PrTime_{uj} \tag{4.23}$$

$$\text{Machine change time:} \quad MCT = \sum_s \sum_j \sum_{j'} cm_{sjj'} \times MCTime_{jj'} \tag{4.24}$$

$$\text{Configuration change time:} \quad CCT = \sum_s \sum_j \sum_l \sum_{l'} cc_{sj}^{lql'q'} \times CCTime_{jll'} \tag{4.25}$$

$$\text{Tool change time:} \quad TCT = \sum_s \sum_j \sum_q \sum_{q'} cc_{sj}^{lql'q'} \times TCTime_{qq'} \tag{4.26}$$

Constraints

The constraints can be classified into two main categories: those that were taken from the original model and been modified to a limited extent (From constraints (4.27) to (4.34)), and those that have been added and do not appear in the original formulation as a result of the removal of triplets (From constraints (4.35) to (4.37)). We will begin by examining the constraints that fall into the first category. These can be found in Touzout and Benyoucef, 2019b, but with the addition of triplets.

$$\sum_j \sum_l \sum_q \sum_u x_{us}^{jlq} = 1 \qquad \forall s{=}1,..,NOP \tag{4.27}$$

$$\sum_j \sum_l \sum_q \sum_s x_{us}^{jlq} = 1 \qquad \forall u{=}1,..,NOP \tag{4.28}$$

$$\sum_j \sum_l \sum_q x_{us}^{jlq} \times OPP_{uu'} \leq \sum_j \sum_l \sum_q \sum_{s'} x_{u's'}^{jlq} \qquad \forall u,u'{=}1,..,NOP, \quad \forall s{=}1,..,NOP \tag{4.29}$$

$$\sum_l \sum_q y_s^{jlq} = 1 \qquad \forall j{=}1,..,m, \quad \forall s{=}1,..,NOP \tag{4.30}$$

$$y_s^{jlq} \geq \sum_u x_{us}^{jlq} \qquad \forall j{=}1,..,m, \quad \forall s{=}1,..,NOP, \quad \forall l{=}1,..,NC_j, \quad \forall q{=}1,..,T \tag{4.31}$$

$$\sum_u \sum_l \sum_q x_{us}^{j'lq} + x_{us-1}^{jlq} \leq cm_{sjj'} + 1 \qquad \forall s{=}2,..,NOP, \quad \forall j,j'{=}1,..,m, \tag{4.32}$$

$$y_s^{jl'q'} + y_{s-1}^{jlq} \leq cc_{sj}^{lql'q'} + 1 \qquad \forall s{=}2,..,NOP, \quad \forall j{=}1,..,m, \quad \forall l,l'{=}1,..,NC_j, \quad \forall q,q'{=}1,..,T \tag{4.33}$$

$$\sum_l \sum_{l'} \sum_q \sum_{t'} cc_{sj}^{lql'q'} = 1 \qquad \forall s{=}1,..,NOP, \quad \forall j{=}1,..,m, \tag{4.34}$$

The second group of constraints that we introduced:

$$x_{us}^{jlq} \times OPTAD_{ut} \leq CTAD_{jlt}$$
$$\forall u{=}1,..,NOP, \quad \forall j{=}1,..,m, \quad \forall s{=}1,..,NOP, \quad \forall l{=}1,..,NC_j, \quad \forall q{=}1,..,T, \quad \forall t{=}1,..,8 \tag{4.35}$$

$$x_{us}^{jlq} \leq MT_{jq} \qquad \forall u{=}1,..,NOP, \quad \forall j{=}1,..,m, \quad \forall s{=}1,..,NOP, \quad \forall l{=}1,..,NC_j, \quad \forall q{=}1,..,T \tag{4.36}$$

$$x_{us}^{jlq} = 0 \qquad \forall u{=}1,..,NOP, \quad \forall j{=}1,..,m, \quad \forall s{=}1,..,NOP, \quad \forall l{=}1,..,NC_j, \quad \forall q{=}1,..,T, \quad q{\neq}OPT_u \tag{4.37}$$

Two shortcomings are evident in the 2MDV modelling approach. Firstly, additional constraints are required to establish the relationship between the two decision variables x and y, despite the fact that all the necessary information is contained in the $x_{i,j}^{k}$ values. 2MDV needs y_s^{jlq} to follow the potential changeovers in machine configuration and tooling between any two positions where the machine is utilized. However, it is unlikely that a machine's configuration or tooling would be changed unnecessarily, particularly when the objective is to minimize total production time. This results in the second disadvantage, the unnecessary tracking of the machines' state in each position, despite the fact that only one machine is utilized at a time, implying that only one machine's state may change in each position. Furthermore, 2MDV does not permit the utilisation of data sets with changeover times that doesn't support the triangle inequality.

4.3.2.1 *Valid inequality for 2MDV*

We propose the valid inequalities (4.38) which effectively reduce the computation time for a tested instance on 2MDV from 70 seconds to 7 seconds. Additionally, they rectify the issue of compatibility with changeover times that do not comply with the triangle inequality. However, in the tests conducted in Section 4.6, we did not include these constraints in 2MDV because we believe it would provide an unfair advantage over 1MDV, where no valid inequalities were used, and would prevent a fair comparison between the two modelling approaches.

$$y_s^{jlq} \leq y_{s-1}^{jlq} + \sum_{u}\sum_{l'}\sum_{q'} x_{us}^{jl'q'} \qquad \forall j{=}1,..,m, \quad \forall s{=}2,..,NOP, \quad \forall l{=}1,..,NC_j, \quad \forall q{=}1,..,T, \tag{4.38}$$

4.3.2.2 *Preliminary comparison 1MDV vs 2MDV*

Table 4.5 presents the comparative results for the smallest instance (M1OP5) in terms of the number of binary variables and constraints for the two models. Further details on the instances can be found in Section 4.6. As can be seen from the table, 1MDV requires fewer variables and constraints than 2MDV. In Section 4.6.2, we evaluate the computational performance of the models in solving 28 different instances.

Table 4.5: Comparing the two models in number of variables and constraints

Model	Binary variables	Constraints
2MDV	12 965	21 401
1MDV	2 162	9 639

4.4 LOWER BOUNDS

The lower bound presented in Mahmoodjanloo et al., 2020 provided useful inspiration for this lower bound model. As the sequencing decisions constitute the primary source of the model's complexity, relaxing these decisions can result in a notable reduction in the overall complexity of the problem. Consequently, in this lower bound model, only machines are assigned to operations with the objective of reducing the total processing time, which results in *LB1*. Next, we expand this model to *LB2*, where machines are arranged in a sequence to minimize the total machine changeover time plus processing time, but similar to LB1, they both don't consider precedence constraints.

Therefore, the strategy is to assign each operation to the machine with the highest processing efficiency for *LB1* (Constraints 4.42-4.46). Subsequently, in *LB2* the objective is to minimise the changeovers required to complete the sequence of machines, as defined by constraints (4.47-4.53).

The TSP and our sequencing problem are similar in that machines are analogous to cities that must be visited a specific number of times to complete the operations assigned to them. In this formulation, each city (machine) j is a potential destination that can be visited from another city (machine) j', in this case $Mch_{j'j} \geq 1$. Also a city may be visited from it ($Mch_{j,j} \geq 1$) implying that the same machine is utilized for two subsequent operations. The objective of the process planner is to determine the optimal sequence of cities (machines) in order to minimize the total distance traveled, which is equivalent to the sum of machines changeover time.

It is important to note that the solution obtained by this method identifies the machine sequence, rather than the exact positions of the machines in the process. So, if $Mch_{13} = 1$ we know that machine 1 is used then immediately followed by machine 3, however we do not know the position of Machine 1 or 3 in the sequence.

The *LB2* mathematical model will be described in detail herein, along with the relevant decision variables. The same notations for indices and parameters previously presented in 1MDV will be used. It should be noted that *LB1* is a sub-part of LB2 model.

The following decision variables are used in the lower bound models.

Decision variables	
ch_u	The minimum value of processing time for operation u
mc_{uj}	Equals 1 if operation u is assigned to machine j, 0 otherwise
$Mch_{jj'}$	Number of machine changeovers from j to j', 0 otherwise
d_j	Equals 1 if machine j is in the first machine used, 0 otherwise
f_j	Equals 1 if machine j is in the last machine used, 0 otherwise
$c_{jj'}$	Equals 1 if machines j and j' are not used to process any operation, 0 otherwise

The ILP formulation for *LB2* calculation follows:
Objective function; (Minimize the processing time and machine changeover time)

$$\text{Minimize: } f_{time} = PT + MCT \tag{4.39}$$

$$\text{Processing time:} \qquad PT = \sum_u ch_u \tag{4.40}$$

$$\text{Machine changeover time:} \qquad MCT = \sum_{jj'} Mch_{jj'} \times MCTime_{jj'} \tag{4.41}$$

Constraints

$$ch_u = PrTime_{uj} + M \times (1 - mc_{uj}) \qquad \forall u \in U, \quad \forall j \in J, \tag{4.42}$$

$$\sum_j mc_{uj} = 1 \qquad \forall u \in U, \tag{4.43}$$

$$\sum_q (OPT_{uq} \times MT_{jq}) \geq mc_{uj} \qquad \forall u \in U, \quad \forall j \in J, \tag{4.44}$$

$$mc_{uj} \leq \frac{1}{\prod_l (-\sum_{tad} OPTAD_{utad} \times (CTAD_{jltad} - 1)) + 1} \qquad \forall u \in U, \forall j \in J, \tag{4.45}$$

$$mc_{uj} \in \{0,1\} \qquad \forall u \in U, \forall j \in J, \tag{4.46}$$

$$\sum_{j'} Mch_{j'j} + d_j = \sum_u mc_{uj} \qquad \forall j \in J, \tag{4.47}$$

$$\sum_{j'} Mch_{jj'} + f_j = \sum_u mc_{uj} \qquad \forall j \in J, \tag{4.48}$$

$$\sum_j d_j = 1 \tag{4.49}$$

$$\sum_j f_j = 1 \tag{4.50}$$

$$(d_j + f_j - \sum_{j'}^{j \neq j'} Mch_{jj'} + Mch_{j'j} - 2) \times TNOP \leq Mch_{jj} - TNOP + 1 \quad \forall j \in J, \tag{4.51}$$

$$c_{jk} \leq 1 - \frac{\sum_u mc_{uj} + mc_{uk}}{TNOP} \qquad \forall j,k \in J, j \neq k \tag{4.52}$$

$$(-c_{jk} - \sum_{j'}^{j' \neq j,k} (Mch_{jj'} + Mch_{j'j}) - \sum_{j'}^{j' \neq j,k} (Mch_{kj'} + Mch_{j'k})) \times TNOP$$
$$\leq Mch_{jj} + Mch_{kk} + Mch_{jk} + Mch_{kj} - TNOP + 1 \qquad \forall j,k \in J, j \neq k \tag{4.53}$$

The objective stated in equation (4.39) is to reduce the production time and machine changeover time, the machine changeover time is stated in equation (4.41) and the processing times of the operations stated in equation (4.40). If operation u is assigned to machine j, constraints (4.42) ensure that ch_u is precisely equal to that processing time. Constraints (4.43) specify that only a single machine is assigned to each operation. Constraints (4.44) and (4.45) guarantee that the necessary resources (tools and TADs) for the operation are available on the specified machine. As stated in Constraints (4.46), the variable x_{uj} is binary. The number of changeovers to a machine must be equal to the number of changeovers that exit the machine. This value must also be equal to the number of operations assigned to that machine, as per constraints (4.47) and (4.48). In order to satisfy the constraints (4.49) and (4.50), one machine is designated as the first and another as the last. Subtours involving a single machine are eliminated by constraints (4.51), whereas subtours involving two machines are eliminated by constraints (4.52) and (4.53). It is crucial to highlight that utilising the "=" symbol in the mathematical model to implement constraints (4.42), (4.43), (4.47), (4.48), (4.49), and (4.50) is not a feasible approach in practice. In order to maintain equivalence, it is necessary to replace each of these constraints into two constraints, with one constraints having a sign of "$\leq$" and the other having a sign of "$\geq$".

As *LB1* is a subproblem of LB2, it focuses solely on reducing the processing time with Constraints (4.42)-(4.46), using just the ch_u and $x_{u,j}$ decision variables.

Tools and TAD constraints (Constraints (4.44) and (4.45)) are included into the LB model. As a result, it can verify whether there's a feasible process plan without taking precedence constraints into consideration[1]. In simpler terms, the model determines if at least one machine is able to process each operation. This fast feasibility check is especially useful in large-scale cases or when computation time is limited.

1 Feasibility here is only in terms TADs and tools constraints, since the model doesn't consider precedence constraints.

4.5 UPPER BOUND (A MATHEMATICAL PROGRAMMING-BASED HEURISTIC)

The 1MDV mathematical model and *LB1*, discussed in sections 4.3.1 and 4.4, are combined in a hierarchical structure to form the heuristic framework proposed in this section. As illustrated in Figure 4.7, the heuristic tackles the problem in three successive hierarchical phases. In the initial phase, the *LB1* model is employed to allocate operations to machines with the objective of minimizing the total processing time. In the second stage, a sequencing problem with solely precedence-related constraints is solved using the 1MDV relaxed model. The objective is to minimize the machine changeover times. In the third phase, the 1MDV relaxed model is utilized to determine the best configurations and tools for the given operations and machines sequence, while accounting for the constraints of the operational requirements (TADs and tools) and without the precedence constraints. As highlighted in Figure 4.7, the values of the decision variable P^{u}_{sjlq} are gradually fixed by the model at each step until they are all known. Furthermore, the objective function is progressively refined by reducing the processing time, the machine changeover time, and configurations and tools changeover times.

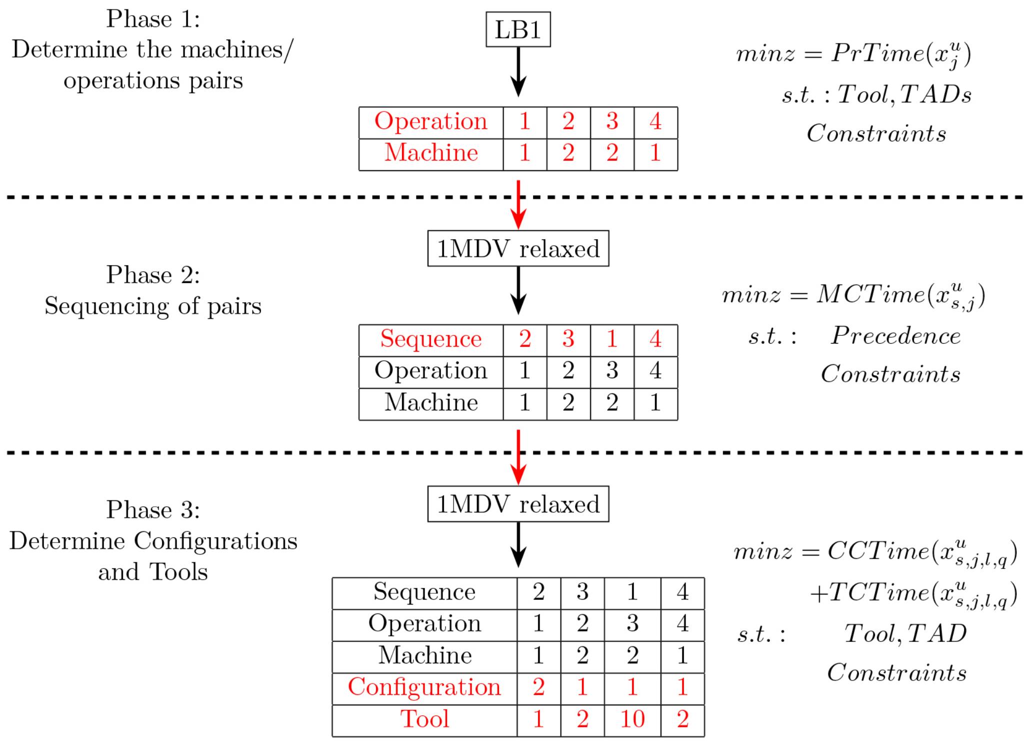


Figure 4.7: Illustration of the three phases of the mathematical based heuristic Mechaacha et al., 2024

After carrying out the three stages illustrated in Figure 4.7, we get at the first version of the heuristic, which we will designate Heuristic *H1.0*. After completing all three phases, we can obtain another variant of this heuristic by repeating the second phase (operations sequencing), where we sequence the quadruplets made up of operations, machines, configurations, and tools. We call this new vari-

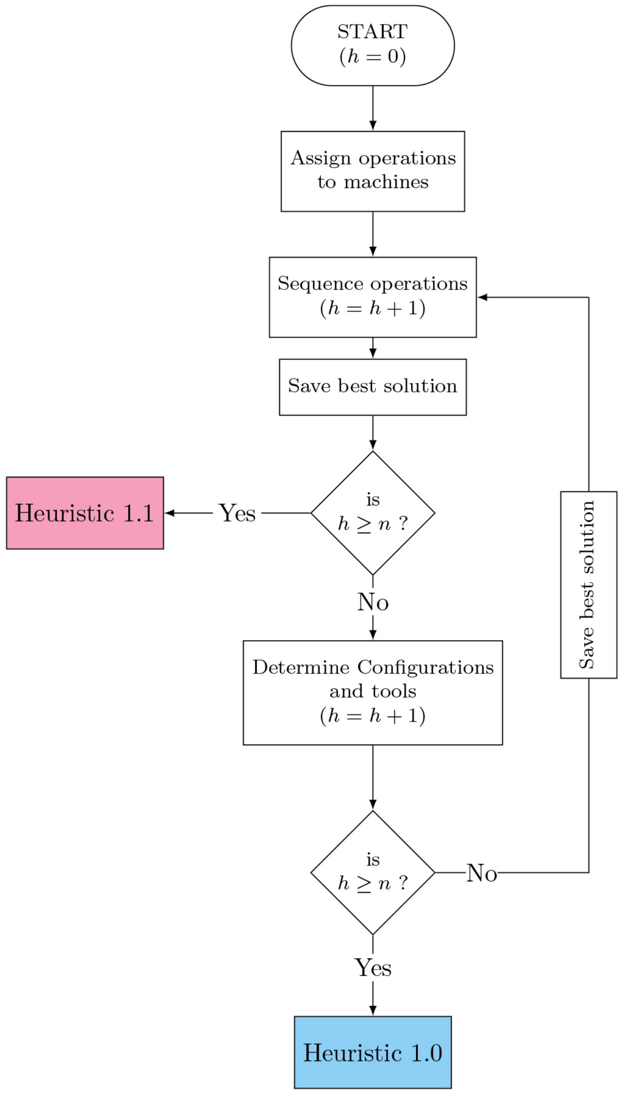


Figure 4.8: Flowchart explaining the two variants of the heuristic H1.0 and H1.1 Mechaacha et al., 2024.

ation Heuristic *H1.1*. Figure 4.8 illustrates the algorithmic representation of both variants. By varying the value of parameter h, we can control the version that is generated. The method produces an H1.0 solution when $h = 2$, and an H1.1 solution when $h = 3$.

It is noteworthy that an increase in the value of h will result in other heuristic alternatives. However, while this may lead to a higher-quality solution, it will also require a greater computing time.

4.5.1 *Data Reuse for New Product Integration*

The fact that RMS is constructed based on a product family, which is defined as a set of products that share similar characteristics, is one of the primary attributes that differentiate it from other systems. In a situation where our heuristic is required to function in such a scenario, it is possible to retain the data produced between stages for other products (Figure 4.7) rather than discarding it. Instead, this data is stored in a knowledge repository along with the final solutions. The repository thus serves as an invaluable tool for process planning during the development of other or new items in the product family.

In order to take advantage of the potential of the heuristic framework, in a semi-generative strategy, the planner has the option of selecting which phase of the heuristic to incorporate the information about the new product. To illustrate, in the case of a new product with additional operations, these will be incorporated into the existing process plan at the lowest level (phase 3 of Figure 4.7) of the heuristic and assigned to the relevant machines, configurations, and tools. In this scenario, only the sequencing of all the operations will be repeated in phase 4 of *H1.1* (Figure 4.8), which will facilitate the implementation of rapid rectifications to the old plan. An alternative strategy would be to incorporate the new product data in phase 2, when the tools and configurations are selected before all operations are sequenced, and so on, ultimately leading to a completely generative method where the data is integrated from the beginning. The desired quality of the solution and the amount of time available will determine the appropriate level of integration for the new product data.

It is important to note that this proposed data reuse has not been subjected to numerical experimentation within the context of this thesis. Consequently, further testing and investigation are needed to assess its effectiveness, which remains an open question in the research domain.

4.6 NUMERICAL EXPERIMENTS

This section presents the various data sets that were evaluated initially. It then provides the computational results of applying the two models, two lower bound computations, and the two heuristic variations. Finally, a discussion of the results is provided, along with managerial insights.

4.6.1 *Data sets*

In the experimental tests, we generate process plans for seven parts to be manufactured on an RMS with $m = 1$, 5, 10, and 20 machines, resulting in 28 distinct instance sizes (7×4). The aforementioned methods (Sections 4.3, 4.4, and 4.5) are employed for this purpose. 5 instances were randomly generated for each combination of m and parts using varying time-based data, resulting in a total of 140 instances (28×5). It is noteworthy that a general-purpose machine with all available functionalities (Tools and TADs) was considered in the case of a single machine, which may not be the most realistic representation of an RMS. It should be noted, however, that this case was taken only for experimental purposes.

The parameters used in this study can be categorized into two groups: those derived from a distribution based on values extracted from existing literature and those obtained directly from other research articles in the same field. The precedence graph, TADs for each operation, the number of operations, and the processing tools required were among the data extracted from the literature. The majority of these are derived from the study by S. Zhang and Wong, 2016. The time-related data, notably the processing and changeover times, constitute the parameters that have been generated, as the study by S. Zhang and Wong, 2016 did not propose an RMS for processing the parts. The number of machines, machine configurations, TADs of configurations, and tools available for each machine are the ones generated randomly. However, these were shaped by the example provided by Haddou Benderbal et al., 2018.

The seven-part data set described in S. Zhang and Wong, 2016 was utilized, which contains details on the requirements of each part in terms of TAD, tool, and precedence graph. The first parts 1 and 2 were illustrated in Chapter 1. The data and figures for the other parts are available in S. Zhang and Wong, 2016. Moreover, the data from parts 5, 6, and 7, as well as the 12 tools, were obtained from W. Li and McMahon, 2007. For further details on the seven parts and the 12 tools, please see the two cited works. It is noteworthy that parts 2 and 4 contain alternative sets of operations that are not permitted in the present study. As a result, the OR branches were removed from the precedence graph, which resulted in the reduction of the number of operations from 13 to 8 for part 2 and from 15 to 13 for part 4, respectively.

In order to obtain time-related data for our research, we replicated the experimental data suggested in Haddou Benderbal et al., 2018 in their instance with 10 machines and 15 operations. A uniform distribution with a range of 100 to 800 was employed to construct the processing time matrix. On the other hand, a hybrid technique was employed to generate the changeover time matrices (configuration and tool) given that the 2MDV model does not support changeover times that do not satisfy the triangle inequality (see Appendix A). Moreover, as the machine changeover times are not required to satisfy the triangle inequality, a uniform distribution with a range between 5 and 64 was employed to generate them.

A square matrix of size n, representing the number of machines, configurations, or tools, constitutes the changeover time matrix. The time required to transition from machine, configuration, or tool i to j is represented by the entry a_{ij}. The configuration changeover times have a mean of 30 and standard deviation of 22, while for tool changeover times, they are 25, 11 for the mean and standard deviation, respectively.

4.6.2 *Experimental Results*

The CPLEX solver, version 12.8.0, was used to run the tests, which were conducted on a laptop equipped with an Intel Core i5 CPU at 1.20 GHz and 8GB of RAM. There were two distinct sets of tests, the initial set of tests was to compare the gaps between the generated lower bounds of *LB1*, *LB2*, 1MDV, and 2MDV and the nearest lower bound to the ideal solution, designated as the "*BestLB*." The objective of the second series of tests was to assess the capacity of the heuristics and models to generate optimal or near-optimal solutions.

4.6.2.1 *Quality of lower bounds*

The four models (*LB1*, *LB2*, 1MDV, and 2MDV) are examined with respect to their capacity to provide lower bounds in terms of CPU times and closeness to the optimal solution. It is easy to see, that 1MDV and 2MDV models are capable of reducing the gap and identifying optimal solutions when provided with sufficient time, while the two bounds run, in a short CPU time, until the lower is found and stop. To ensure a fair comparison between these two types of models, we conducted tests in which *LB1*, and *LB2* were permitted to run freely, and the 1MDV and 2MDV models were given sufficient time to reach the same lower bounds as *LB1*. This approach allowed us to evaluate the time required for the models to achieve results that were of the same quality as *LB1*.

It should be noted that the stopping criterion was set as a percentage of the optimality gap. This was determined after preliminary tests were conducted to obtain lower bounds that were close to LB1 values. This approach was necessary because CPLEX does not directly support stopping criteria based on reaching a specific lower bound value. For all instances, the optimality gap was set to 90%. The only exception was the instance M1P1(OP5), where the gap was set at 50%. The gaps to the BestLB were calculated using the following formula:

$$\text{Gap of LB from BestLB} = \frac{\text{BestLB} - \text{LB}}{\text{BestLB}} \times 100$$

The results of the lower bounds test are presented in Table 4.6. In addition to the CPU time, we also present the gap from BestLB for each method utilized to generate lower bounds. The first column lists the instance codes, which represent the number of machines, the component number, and the total number of operations of the part. To illustrate, instance M1P1(OP5), comprising a single machine and five operations, with part number 1 as the manufactured part.

The lower bounds obtained by the LB1, 2MDV, and 1MDV models are approximately 2% lower than the BestLB, indicating a similar performance. As previously stated, the decision to utilize the stopping criterion for the two models is the primary reason for this proximity. Conversely, the LB1 model is considerably more rapid than the 1MDV and 2MDV models (28 and 86 seconds, respectively, in comparison to 0.03 seconds averages). In the majority of instances, LB2 is the best lower bound. The mean distance between LB2 and bestLB is 0.08%, and in 93% of instances, LB2 is the better bound. Although LB2 has a computational time of less than half a second in the most challenging instance and an average of 0.15 seconds, it is somewhat slower than LB1. This challenging instance is, as expected, the largest with 20 machines and 20 Operations.

The results demonstrate that LB2 is the optimal choice for implementation in a Branch and Bound framework or for assessing the quality of heuristics.

Table 4.6: The CPU time (in seconds) and the gaps (in percent) between the obtained lower bounds and the best known lower bounds (the 1MDV and 2MDV models were stopped when their lower bounds approached LB1).

Instance	LB2		LB1		2MDV Model		1MDV Model	
	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)
M1P1(OP5)	1.81	0.01	1.81	0.02	0	0.14	0	0.14
M1P2(OP8)	0	0.01	0	0.01	0	0.32	0	0.45
M1P3(OP6)	0	0.01	0	0.01	0	0.17	0	0.22
M1P4(OP13)	0	0.01	0	0.01	0	0.96	0	1.21
M1P5(OP20)	0	0.01	0	0.01	0	3.58	0	5.12
M1P6(OP16)	0	0.01	0	0.01	0	1.74	0	2.66
M1P7(OP14)	0	0.01	0	0.01	0	1.21	0	1.84
M5P1(OP5)	0	0.02	1.41	0.01	0.95	1.47	0.9	0.5
M5P2(OP8)	0	0.02	1.41	0.01	1.31	2.86	1.31	2.18
M5P3(OP6)	0	0.03	2.74	0.01	2.68	1.79	2.68	0.85
M5P4(OP13)	0	0.04	1.59	0.02	1.59	15.5	1.59	10.26
M5P5(OP20)	0	0.03	0.78	0.02	0.78	30.31	0.78	49.84
M5P6(OP16)	0	0.02	0.93	0.02	0.93	21.25	0.93	22.58
M5P7(OP14)	0.51	0.02	1.07	0.01	0	14.33	0.07	14.5
M10P1(OP5)	0	0.05	4.79	0.02	4.44	2.68	4.44	1.57
M10P2(OP8)	0	0.07	2.28	0.02	2.17	9.27	2.12	5.54
M10P3(OP6)	0	0.08	3.69	0.02	3.69	2.91	3.69	2.12
M10P4(OP13)	0	0.1	2.63	0.03	2.63	15.54	2.63	21.76
M10P5(OP20)	0	0.09	1.33	0.04	1.33	299	1.33	99.8
M10P6(OP16)	0	0.08	1.23	0.03	1.23	39.98	1.23	44.73
M10P7(OP14)	0	0.07	1.4	0.03	0.94	43.96	0.93	28.38
M20P1(OP5)	0	0.4	7.79	0.03	7.12	5.28	7.13	2.5
M20P2(OP8)	0	0.56	3.94	0.04	3.94	56.9	3.94	11.28
M20P3(OP6)	0	0.47	6.33	0.04	6.33	6.07	6.33	4.54
M20P4(OP13)	0	0.44	4.17	0.06	4.17	237	4.17	46.86
M20P5(OP20)	0	0.5	2.32	0.07	2.32	853	2.32	263
M20P6(OP16)	0	0.55	2.53	0.05	2.53	358	2.42	92.9
M20P7(OP14)	0	0.33	2.68	0.05	2.4	405	2.68	61.7
AVERAGE	**0.08**	**0.15**	**2.11**	**0.03**	**1.91**	**86.89**	**1.92**	**28.57**

The approach used to evaluate the lower bounds, which involved setting a stopping criterion of matching LB1 values, may be viewed as an uncommon methodology for evaluation by the reader. Nevertheless, its necessity is evident, at least from our perspective. To illustrate this necessity, Table 4.7 presents the averages recorded from setting different stopping criteria, including three time limits (2, 100, and 900 seconds) and the previously presented criterion. As it can be seen from Table 4.7, CPU time of LB1 and LB2 is never influenced by the time limit for any criterion, as both lower bounds remain below the two-second threshold for all instances. In the context of the 2 seconds limit, the models were unable to generate a lower bound for the majority of instances. This prevents the calculation of averages and, consequently, a meaningful comparison. Upon increasing

Table 4.7: Averages of CPU times (in Seconds) and gap (in %) of obtained lower bounds from best known lower bounds for different stopping criteria.

Stop criterion	LB2		LB1		2MDV Model		1MDV Model	
	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)
2 sec Limit	**1.05**	**0.15**	**2.97**	**0.03**	**-**	**-**	**-**	**-**
100 sec Limit	**2.02**	**0.15**	**3.92**	**0.03**	**4.87***	**60.67**	**3.90***	**43.73**
900 sec Limit	**2.34**	**0.15**	**4.31**	**0.03**	**1.13**	**451.21**	**0.17**	**328.69**
Achieve LB1 value	**0.08**	**0.15**	**2.11**	**0.03**	**1.91**	**86.89**	**1.92**	**28.57**

* The instances where the models couldn't find any lower bound within 100 seconds were set as 100% gap.

the limit to 100 seconds, 1MDV and 2MDV generate lower bounds for the majority of instances. However, in the case of 1MDV, one instance was unsolved, while in the case of 2MDV, two instances displayed this behavior. In these instances, the gaps were replaced with 100%. This value has affected the calculation of averages in a way that is misleading. It has caused the averages to increase, which gives the impression that the two models were generating poor-quality lower bounds. However, the majority of the lower bounds generated by the models were of good quality. Finally, as the tests of the following section on heuristics are conducted with a time limit of 900 seconds, we tried setting the limit to 900 seconds. In this case the mathematical model had an unfair advantage, with 1MDV having 0.17% gap and LB2 being able to surpass it in only very few large instances.

4.6.2.2 *Advanced analysis on Lower bounds*

In the preceding subsection, it was shown that the choice of stopping criteria affects the generation of Lower bounds. In this subsection, we conduct a comprehensive analysis of the results obtained from the two stopping criteria: "*achieve LB1 value*" and "*900-second limit*." In particular, we examine how the variation in the number of machines and operations affects the variation in gaps and CPU time.

Table 4.8 depicts the gap between lower bounds and the best lower bound per number of machines and operations, as well as the average computational time for the "*achieve LB1 value*" case. Table 4.9 presents the results of the "*900 sec limit*," in which the models were allowed 900 seconds of CPU.

As can be seen in Table 4.8, the CPU time taken to calculate the lower bounds is relatively unaffected by an increase in the number of operations. In contrast, the two models demonstrate an exponential increase in CPU time as the number of operations and machines increase. Table 4.8 does not provide significant insight into the gaps, as the 1MDV, 2MDV, and LB1 results are similar, whereas LB2 is consistently the BestLB.

Table 4.9 reveals that the average gap between LB1 and LB2 and BestLB decreases as the number of operations increases. This suggests that the lower bound

Table 4.8: Averages per number of machines and operations for lower bounds with the stopping criterion: *Achieve LB1 value*

Number of	LB2		LB1		2MDV Model		1MDV Model	
Machines	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)
1	0.26	0.01	0.26	0.01	0	1.17	0	1.67
5	0.07	0.03	1.42	0.02	1.18	12.51	1.18	14.39
10	0	0.08	2.48	0.03	2.35	59.06	2.34	29.15
20	0	0.47	4.26	0.05	4.12	274.51	4.15	69.10
Operations								
5	0.45	0.13	3.95	0.03	3.13	2.40	3.12	1.18
6	0	0.15	3.19	0.02	3.18	2.74	3.18	1.94
8	0	0.17	1.91	0.03	1.86	17.36	1.85	4.87
13	0	0.15	2.10	0.03	2.10	67.46	2.10	20.03
14	0.13	0.11	1.29	0.03	0.84	116.35	0.92	26.63
16	0	0.17	1.18	0.03	1.18	105.45	1.15	40.74
20	0	0.16	1.11	0.04	1.11	295.92	1.11	104.64

Table 4.9: Averages per number of machines and operations for lower bounds with the stopping criterion: *900 sec limit*

Number of	LB2		LB1		2MDV Model		1MDV Model	
Machines	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)
1	1.75	0.01	1.75	0.01	0.03	107.37	0	63.82
5	2.67	0.03	3.98	0.02	0.98	504.33	0.01	362.46
10	2.38	0.08	4.79	0.03	1.26	558.96	0.12	396.84
20	2.57	0.47	6.69	0.05	2.26	634.17	0.56	491.64
Operations								
5	4.67	0.13	7.97	0.03	0	6.42	0	1.15
6	1.87	0.15	5.00	0.02	0	28.27	0	2.43
8	3.09	0.17	4.94	0.03	0.10	217.62	0	8.40
13	1.71	0.15	3.79	0.03	3.13	686.32	0.20	593.51
14	4.00	0.11	5.10	0.03	2.31	626.80	0	158.03
16	0.55	0.17	1.73	0.03	1.16	700.92	0.42	707.40
20	0.51	0.16	1.62	0.04	1.21	892.11	0.58	829.93

model exhibits enhanced performance when the number of operations is high. Nevertheless, it could be argued that this convergence is due to a deterioration in the quality of BestLB, which is often equal to the 1MDV LB, as the 1MDV model frequently reaches the 900-second limit in 20 operation instances. To address this concern, an analysis of the machine-related gap on the same table (4.9) reveals an opposite behavior, with an increase in the lower bounds gap as the number of machines increases, particularly for 20 machines, with LB2 exhibiting a lesser increase. Based on these observations, it can be concluded that the lower bound model exhibits superior performance when the number of operations is high. This may be attributed to the fact that the proportion of processing time in the total time increases as the number of operations rises, given that the lower bounds are designed to optimise processing time primarily.

4.6.2.3 *Performance of heuristics*

In order to assess the relative quality of the solutions produced by the two models (1MDV and 2MDV) and the two heuristic variants (H1.0 and H1.1), the second set of tests was conducted with a maximum CPU time of 900 seconds. The time limit was set on the basis of the literature (see, for instance, Khezri et al. 2021; Touzout et al. 2018). The evaluation is based on the gap obtained between the best lower bound (among LB2 and the lower bounds of the 1MDV and 2MDV models), and the best solution found using a particular method. Of course, for the instances where the integrality gap of the mathematical models was closed, their best lower bound is automatically equal to the optimal solution value. Table 4.10 presents the full results of these tests, whereas the gaps are calculated as follows:

$$\text{BestLB } (\%) = \frac{\text{Solution} - \text{BestLB}}{\text{BestLB}} \times 100$$

$$\text{BestSol } (\%) = \frac{\text{Solution} - \text{Best Found Solution}}{\text{Best Found Solution}} \times 100$$

Table 4.10: Comparing CPU times (in Seconds) and gaps (in %) between the solutions obtained and the best lower bounds, and between the the solutions obtained and the best solution discovered.

Instance	2MDV Model				1MDV Model				H1.0				H1.1			
	Opt.	BestLB (%)	BestSol (%)	CPU (sec)	Opt.	BestLB (%)	BestSol (%)	CPU (sec)	Opt.	BestLB (%)	BestSol (%)	CPU (sec)	Opt.	BestLB (%)	BestSol (%)	CPU (sec)
M1P1(OP5)	5	0	0	0.18	5	0	0	0.35	5	0	0	0.21	5	0	0	0.42
M1P2(OP8)	5	0	0	0.51	5	0	0	0.51	0	2.16	2.16	0.26	3	1.08	1.08	0.59
M1P3(OP6)	5	0	0	0.37	5	0	0	0.32	0	0.57	0.57	0.14	4	0.06	0.06	0.30
M1P4(OP13)	5	0	0	8.18	5	0	0	4.86	0	1.33	1.33	0.98	3	0.18	0.18	4.18
M1P5(OP20)	3	0*	0*	691.71	5	0	0	401.16	0	1.58	1.58	3.07	2	0.15	0.15	99.86
M1P6(OP16)	5	0	0	42.98	5	0	0	33.95	0	1.44	1.43	1.54	0	0.18	0.17	7.22
M1P7(OP14)	5	0	0	7.68	5	0	0	5.59	0	1.24	1.24	1.08	0	0.52	0.52	3.63
M5P1(OP5)	5	0	0	5.26	5	0	0	0.67	3	0.08	0.08	0.50	5	0	0	1.03
M5P2(OP8)	5	0	0	89.60	5	0	0	5.26	1	0.37	0.37	1.65	3	0.06	0.06	3.52
M5P3(OP6)	5	0	0	16.08	5	0	0	1.38	3	0.23	0.23	0.84	4	0.13	0.13	1.62
M5P4(OP13)	0	2.98	1.68	900<	2	1.27	0	622.65	1	1.81	0.54	7.42	1	1.44	0.17	69.80
M5P5(OP20)	0	6.97	3.35	900<	0	4.01	0.49	900<	0	4.37	0.84	33.28	0	3.62	0.11	864.12
M5P6(OP16)	0	3.55	1.05	900<	0	2.48	0	900<	0	3.70	1.19	14.26	0	2.56	0.08	368.61
M5P7(OP14)	2	0.06	0.06	691.66	5	0	0	68.13	0	1.34	1.34	8.38	0	0.84	0.83	31.05
M10P1(OP5)	5	0	0	4.32	5	0	0	1.20	3	0.06	0.06	1.04	4	0.03	0.03	2.05
M10P2(OP8)	5	0	0	232.11	5	0	0	10.11	1	0.99	0.98	3.35	2	0.69	0.69	7.04
M10P3(OP6)	5	0	0	30.95	5	0	0	2.26	2	0.58	0.58	1.61	3	0.49	0.49	3.08
M10P4(OP13)	0	7.29	3.96	900<	2	3.39	0.17	820.91	2	3.43	0.21	15.30	2	3.32	0.10	86.59
M10P5(OP20)	0	19.32	11.18	900<	0	9.73	2.24	900<	0	7.82	0.45	111.02	0	7.34	0	900<
M10P6(OP16)	0	5.49	2.56	900<	0	2.94	0.07	900<	0	4.68	1.77	28.72	0	3.05	0.18	475.44
M10P7(OP14)	1	0.52	0.52	883.72	5	0	0	66.25	0	0.98	0.98	17.80	1	0.21	0.21	46.68
M20P1(OP5)	5	0	0	15.93	5	0	0	2.37	3	1.36	1.36	2.01	3	1.36	1.36	4.08
M20P2(OP8)	4	0	0	548.25	5	0	0	17.73	0	1.92	1.91	6.96	0	1.85	1.84	14.71
M20P3(OP6)	5	0	0	65.69	5	0	0	5.78	2	0.74	0.74	3.68	2	0.66	0.66	6.96
M20P4(OP13)	0	22.96	16.70	900<	0	5.50	0.20	900<	0	6.01	0.67	33.03	0	5.40	0.11	100.14
M20P5(OP20)	0	261.40	241.85	900<	0	8.20	1.98	900<	0	6.38	0.26	194.76	0	6.11	0.00	900<
M20P6(OP16)	0	14.49	9.28	900<	0	4.92	0.13	900<	0	6.28	1.43	61.55	0	5.26	0.46	799.91
M20P7(OP14)	0	8.55	8.55	900<	5	0	0.00	492.14	0	1.04	1.04	36.96	1	0.53	0.53	97.47
	80	**12.63**	**10.74**	**450.74**	**99**	**1.52**	**0.19**	**328.69**	**26**	**2.23**	**0.90**	**21.12**	**48**	**1.68**	**0.36**	**182.67**

* The model does not close the integrality gap for two out of the five instances.

Table 4.10 displays the average gap between the found solution and bestLB, and the average gap to the best discovered solution, each instance size, the number of instances out of the five tested that are solved to optimality, and the average CPU time for each of the four solution approaches.

A detailed examination of the results presented in Table 4.10 allows for the drawing of several observations. Firstly, the 1MDV model outperforms the 2MDV model in all metrics in almost all instances, reaching optimality in 99 of the 140 solved instances, in comparison to the 2MDV model, which only discovered 80 optimal solutions. The mean computational time per instance was 328 seconds for 1MDV, with an average gap to the best LB of 1.52% and an average gap to the best solution of 0.19%. With 1MDV solutions being the best ones, across 21 instance size out of the 28 tested. On the other hand, for 16 instance sizes 2MDV matched the best solution.

When reaching optimality, the 80 instances of both models exhibited notable differences in computational time. The average time for 2MDV was 95 seconds, while 1MDV required just 20 seconds. This indicates an impressive 4.7-fold increase in computational efficiency for 1MDV over 2MDV. However, it is noteworthy that for the smallest instances of M1P1(OP5), 2MDV was able to find the optimal solution in 0.18 seconds, which is faster than 1MDV (0.35 seconds) and H1.0 (0.21 seconds).

The two heuristic variants, H1.0 and H1.1, offer a good balance between CPU time and solution quality. H1.0 demonstrates superior performance to 1MDV in terms of solution quality within a shorter CPU time for the two largest instance sizes. The average computational time for H1.0 is 21 seconds, with an average gap of 0.9% to the best found solution and 2.23% to the best lower bound. Nevertheless, H1.1 solutions demonstrate superior performance to it in terms of solution quality, with an increased average CPU time of 183 seconds.

For the two heuristic variants H1.0 and H1.1, there are a total of 26 and 48 reported optimal solutions, respectively. It is difficult to verify whether optimal solutions were found in cases where the integrality gap was not closed by the 1MDV or 2MDV, which leads us to the known characteristic of heuristics, the inability to verify whether the obtained solutions are optimal.

The advantage of proposing multiple approaches-1MDV, H1.0, and H1.1-is in the ability to adapt to different decision making scenarios, which is shown in Figure 4.9. These methods can be used as a guide for a choice of the best strategy, taking into account the tradeoff between CPU time and solution quality. In an extreme scenario, the decision maker might choose to run the 1MDV model for whatever required amount of time, to obtain a high quality solution. In another scenario, where time is extremely limited, H1.0 might be chosen. Finally, H1.1 would be an intermediate option.

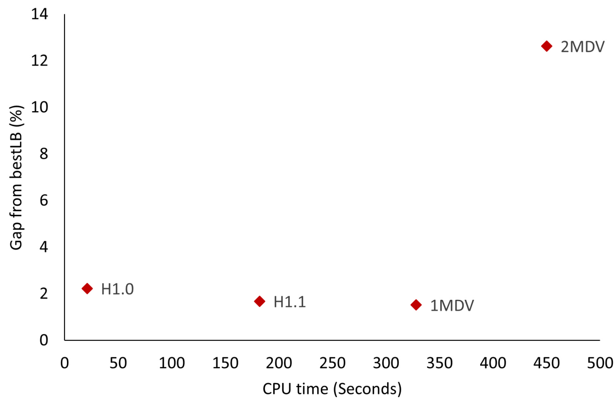


Figure 4.9: CPU times (in seconds) and gaps from bestLB of the four approaches.

Figure 4.10 shows the computation times of the models for the 28 instance sizes, ordered from smallest (M1OP5) to largest (M20OP20), within a the limit of 900 seconds. It is evident that as the instance size increases, the CPU time increases dramatically. Furthermore, the 1MDV model takes less time than 2MDV in every instance, and with the bound of 200 seconds, the heuristic model H1.0 solves all instances.

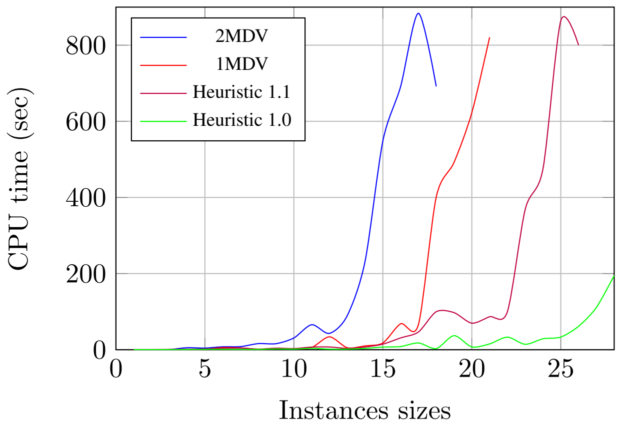


Figure 4.10: The CPU time of the four models for different sized instances ranked from the smallest to biggest.

4.6.2.4 *Advanced analysis on performance of heuristics*

Table 4.11 presents the average CPU times and gaps for all instances of the same number of machines in the first part and instances of the same number of operations in the second part. Figures 4.11, 4.12, 4.13, and 4.14 illustrate the same data in plots, facilitating a deeper comprehension of the methods' behaviour and performance.

A comparison of the results of the two heuristics with 1MDV reveals that 1MDV outperforms H1.0 in all cases, with the sole exception of a single case of 20 operations. In this instance, H1.1 emerges as the superior performer also. Given that the

Table 4.11: Effect of the increased number of machines and operations on the CPU times and solution gaps to best lower bounds for 2MDV, 1MDV, H1.0, and H1.1.

Number of	2MDV		1MDV		H1.0		H1.1	
Machines	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)	Gap from Best LB (%)	CPU (sec)
1	0	107.37	0	63.82	1.19	1.04	0.31	16.60
5	1.94	504.33	1.11	362.46	1.70	9.47	1.24	191.39
10	4.66	558.96	2.29	396.84	2.65	25.55	2.16	225.72
20	43.91	632.28	2.66	491.64	3.39	48.42	3.02	296.95
Operations								
5	0	6.42	0	1.15	0.38	0.94	0.35	1.89
6	0	28.27	0	2.43	0.53	1.57	0.34	2.99
8	0	217.62	0	8.40	1.36	3.05	0.92	6.46
13	8.31	686.32	2.54	593.51	3.14	14.18	2.59	65.18
14	2.28	626.77	0	158.03	1.15	16.05	0.53	44.71
16	5.88	700.29	2.58	707.40	4.02	26.52	2.76	412.79
20	71.92	889.47	5.49	829.93	5.04	85.53	4.31	744.64

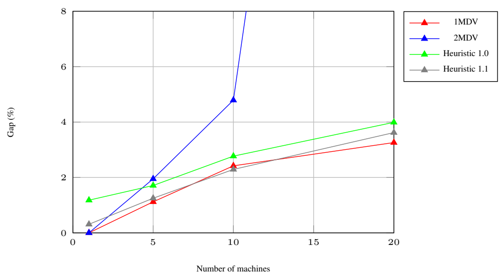


Figure 4.11: Gap of best solutions found to best lower bound per number of machines

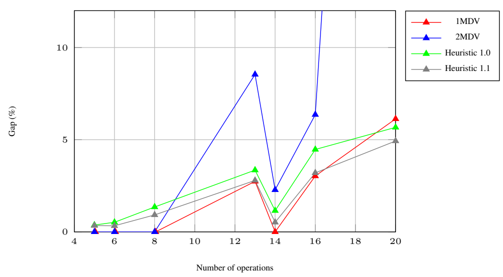


Figure 4.12: Gap of best solutions found to best lower bound per number of operations

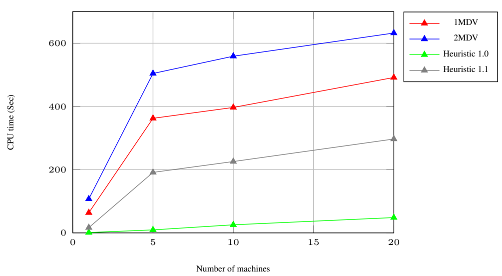


Figure 4.13: CPU time per number of machines

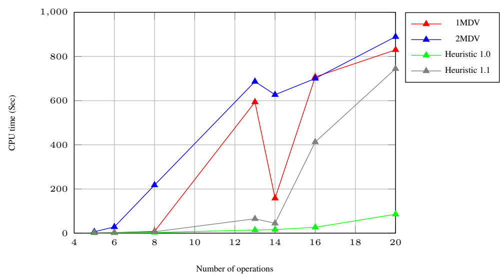


Figure 4.14: CPU time per number of operations

heuristic H1.1 performance tends to converge towards 1MDV's as the number of operations increasesnearly matching it for 16 operations, then surpassing it for the largest configuration with 20 operationsit seems inconsistent that H1.1 performs better than 1MDV for configurations with 10 machines but not in 20.

As can be seen from Table 4.10, there is a direct relationship between the computational times of the four models and the size of the instances. Consequently, larger computational times are the result of adding more machines or operations. Table 4.11 and Figures 4.13 and 4.14 offers further confirmation of this conclusion, indicating also that the number of operations has a bigger impact on the problem's complexity, in comparison to the number of machines.

The results on Table 4.11 demonstrate that as the number of processes and machines increases, there is a clear reduction in the quality of the solutions for all models and heuristics. This deterioration in quality can be attributed to the fact that the models are more likely to exceed the 900-second time limit when dealing with larger instances, resulting in the computing time approaching the limit of 900 seconds. It could be argued that the divergence of the best lower bound, which is often obtained by the 1MDV model and tends to diminish when the model is permitted just 900 seconds, is also responsible for the increase in gap. However, as LB2 has superior lower bounds, it tends to reduce this discrepancy for large instances, as evidenced by surpassing 1MDV bounds in 24 out of the 140 tested instances, all of which were of a large size.

Moreover, H1.1 demonstrates superior performance compared to all H1.0 solutions, and outperforming 1MDV in scenarios involving 10 or more machines and a greater number of operations than 14.

4.6.3 *Insights*

4.6.3.1 *Insights on the Structure of process plans*

The figures presented in 4.15 to 4.20 illustrate the number of changeovers (machines, configurations, and tools) in the optimal solutions obtained by the models per the number of operations and machines. The quality of the solutions is directly influenced by the number of changeovers, with the 1MDV model demonstrating the optimal results and the fewest number of changeovers. The two heuristic variants exhibit a comparable number of machine changeovers to that of the 1MDV, however, the number of tool changeovers is significantly higher due to the prioritization of minimizing machine changeovers. In contrast, the 2MDV model exhibits a slow convergence to optimal solutions, resulting in a significant number of non-optimal solutions, and thus a considerable number of changeovers. Furthermore, as illustrated in Figure 4.20, the increase in the number of operations is associated with a notable rise in the number of tool changeovers, whereas the increase in the number of machines has an opposite effect.

As illustrated in Figures 4.21 and 4.22, the number of machines selected is found to have a positive impact on the quality of the solutions, in contrast to the case of changeovers. The average number of machines selected for 1MDV solutions is 5.1, which represents the highest observed value. This finding suggests that solutions with superior quality tend to have a greater number of machines selected. This trend can be attributed to two factors. Firstly, the selection of machines does not have a direct impact on the total time, as no time is assigned to machine selection. Secondly, selecting more machines can lead to fewer configuration and tool changeovers while still having more machine changeovers. It can also be concluded that an increase in the number of machines or operations generally

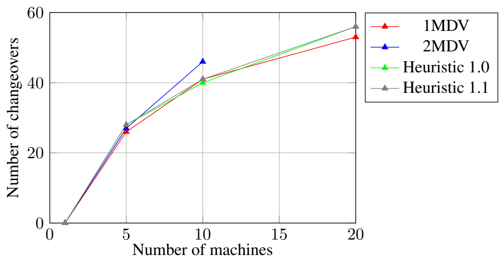


Figure 4.15: Sum of machine changeovers per number of machines

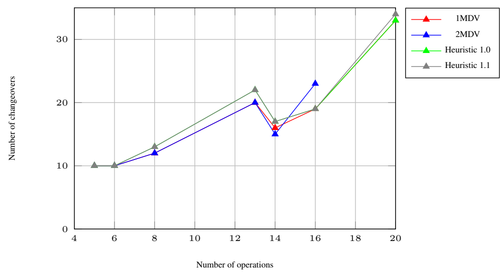


Figure 4.16: Sum of machine changeovers per number of operations

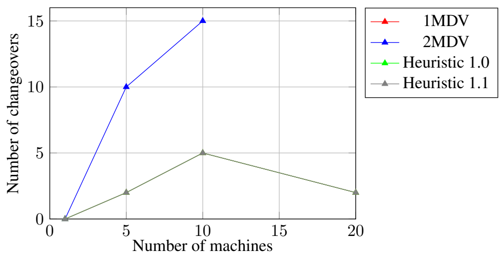


Figure 4.17: Sum of configuration changeovers per number of machines.

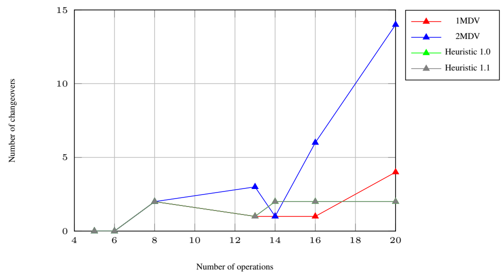


Figure 4.18: Sum of configuration changeovers per number of operations

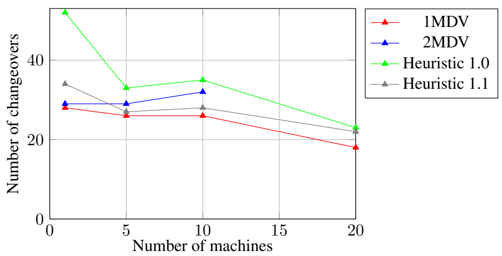


Figure 4.19: Sum of tool changeovers per number of machines

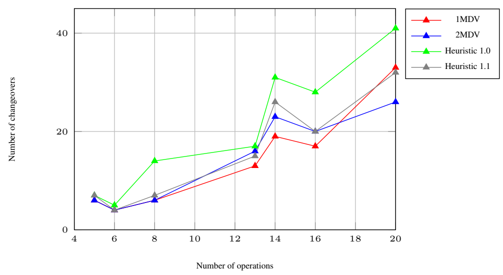


Figure 4.20: sum of tool changeovers per number of operations

results in a greater number of machines being selected in optimal or near-optimal solutions.

An examination of the H1.0 performance as illustrated in Figures 4.21 and 4.22 reveals that H1.0 has a near-optimal selection of machines, although the sequencing is suboptimal. This is evidenced by the numerous changeovers observed between Figures 4.15 and 4.20. The mean absolute value of the difference between the number of machines in the heuristic and the 1MDV model was 0.18, while the value for the difference between the 1MDV and 2MDV models was 0.22. It was thus evident that the sequencing phase should be repeated without alteration to the quadruplets (operation, machine, configuration, tool), which was the motive for us to develop the H1.1 variant.

In most of the figures illustrating the variation per number of operations, an irregular pattern can be observed at 14 operations, which corresponds to part 7. Specifically, there is an unexpected increase in the objective function (Figure 4.23),

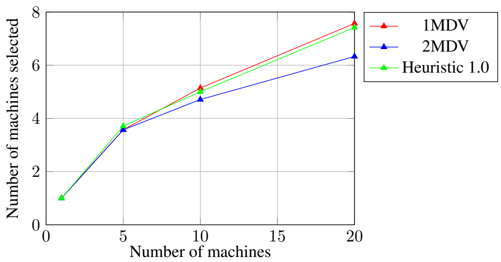


Figure 4.21: Average number of machines selected in a process plan per number of machines

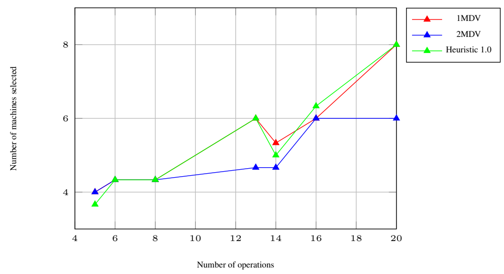


Figure 4.22: Average number of machines selected in a process plan per number of operations

a decrease in CPU time (Figure 4.14), a decrease in the gap to best lower bound (Figure 4.12), and a decrease in the number of selected machines and machine changeovers, along with an increase in the number of tool changeovers (Figures 4.21, 4.16, and 4.20). This suggests that the problem can be more easily solved with fewer machines selected than is typical, and that prioritizing machine changeovers results in a reduction in machine changes but an increase in tool changeovers.

To further investigate this observation, a comparison of the data from operations 13 and 14, corresponding to parts 4 and 7, respectively, was conducted. These parts exhibit opposite behaviors, making them ideal for comparison. The analysis demonstrates that part 7 has a highly constrained precedence graph, which results in a reduction in feasible sequences. Furthermore, the operations necessitate a greater number of TADs than those in part 4, which leads to a smaller number of machines being capable of processing them. As a result, and due to the constraints imposed by the precedence graph, a significant number of tool changeovers are unavoidable. Therefore, it can be concluded that factors other than the instance size play a significant role in determining the complexity of the problem, including the number of relationships in the precedence constraints and the TADs required for processing the operations.

4.6.3.2 *Insights on Total Production Time Variation*

Figures 4.23 and 4.24 illustrate the objective function (total production time) across varying numbers of machines and operations. Figure 4.23 presents the data in three-dimensional ribbons, whereas Figure 4.24 depicts the data from a slightly different angle and displays only the points.

It is evident that the increase in the number of operations and machines gives opposite effects on the total production time. The addition of processing machines to the production workshop results in a reduction in overall production time, primarily due to the decrease in the number of changeovers required. This reduction in production time persists even when the number of machines available exceeds the number of operations. To illustrate, when processing part 1 with 10 machines, the average production time is 1344 (t.u), and it reduces to 962 (t.u) when the number of machines increases to 20, despite the number of operations of the part being only 5. Therefore, increasing the number of machines increases the probability of reducing the total production time by having more machines capable of processing the parts regardless of the number of operations. Conversely, an increase in the number of operations has a more considerable impact on the extension of the production time. This is due to the fact that complex parts require a greater number of operations for processing, which consequently results in an augmented production time.

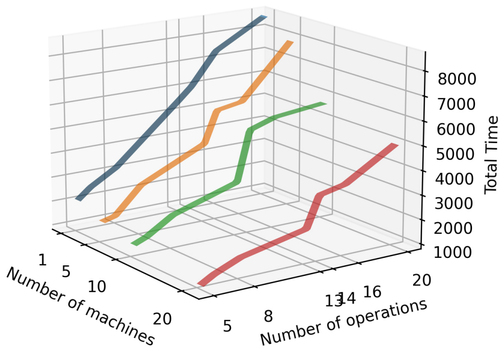


Figure 4.23: The average value of total production time vs. number of machines and number of operations Mechaacha et al., 2024.

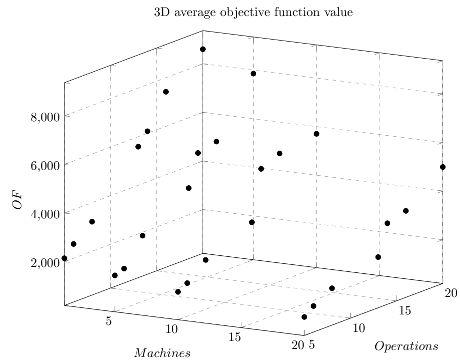


Figure 4.24: The average value of total production time vs. number of machines and number of operations

(a) Best solution found for instance 1 M10P2(OP8) by 1MDV model (TotTime = 2132)

Operation	OP3	OP1	OP2	OP6	OP7	OP4	OP5	OP8
Machine	M4	M10	M10	M10	M10	M1	M1	M1
Configuration	C1	C2	C2	C1	C2	C3	C3	C1
Tool	T8	T6	T6	T6	T1	T4	T9	T9

(b) Best solution found for instance 1 M10P2(OP8) by H1.0 (TotTime = 2150)

Operation	OP3	OP1	OP2	OP6	OP7	OP4	OP8	OP5
Machine	M4	M10	M10	M10	M10	M1	M1	M1
Configuration	C1	C2	C2	C1	C2	C3	C1	C3
Tool	T8	T6	T6	T6	T1	T4	T9	T9

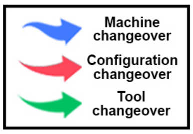


Figure 4.25: The best solutions found for instance 1 M10P2(OP8) by 1MDV and H1.0 Mechaacha et al., 2024.

Solutions (a) and (b) found for instance 1 of M10P2(OP8) using the 1MDV and H1.0 models, respectively, are shown in Figure 4.25. Three machines (M1, M4, and M10) are selected in solution (a), which is the best result produced by 1MDV. This solution and H1.0's reinforces the heuristic's approach of grouping processes assigned to the same machine in a sequential order and then reducing configuration and tool changeovers as much as possible. The machine, configuration, and tool indices would be grouped and replaced by a single index if triplets were used, as in some work in the literature (see, Haddou Benderbal et al. 2018; Khezri et al. 2021; Touzout and Benyoucef 2019b). This would simplify the solution representation, but would require reorganization of the data to track the changeovers.

Two machine changeoversfrom M4 to M10 and from M10 to M1occur in solution (a). Furthermore, M10 is reconfigured twice. If OP2 had been replaced with OP6 or

OP6 with OP7, one reconfiguration may have been avoided. However, these actions had to be carried out in that sequence due to precedence constraints. Additionally, there are two configuration changeovers.

The heuristic H1.0 generated solution (b), which further highlights the usefulness of H1.1 in enhancing H1.0 solutions. The total duration for solution (b) is 2150, whereas the 1MDV solution requires 2132. The H1.1 solution would address the need to swap operations OP5 and OP8 in order to avoid making an additional configuration change, although the solution represented by option (b) is nearly optimal. While the H1.1 technique proved effective in reducing the overall time, it may not be the best option if changeover periods constitute a larger portion of the overall production time.

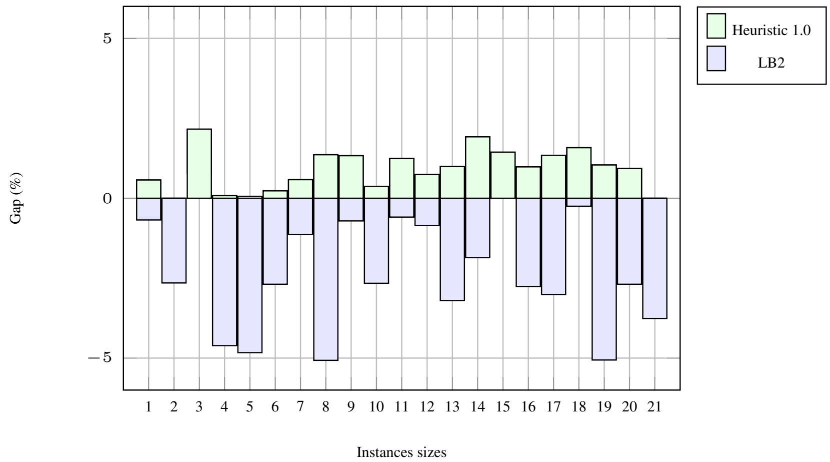


Figure 4.26: The gap percentage between the Upper and Lower bounds to the optimal solution for the 21 problem sizes solved to optimality.

The gaps between LB2 and H1.0 solutions from the optimal values of the 21 problem sizes where the 1MDV model solved at least one of the five instances to optimality are illustrated in Figure 4.26. The results of LB2 and H1.0 can be employed as a lower and upper bound, respectively, as the gaps for the two do not exceed ± 6%. Nevertheless, LB2 does not exhibit the same performance as the upper bound, with an average gap of -3%, in comparison to the UB's average gap of +1.19%. It is also noteworthy that the figure demonstrates that the UB-LB gaps do not undergo a significant deterioration when the instances' sizes increase, from M1OP5 to M10OP14.

4.7 CONCLUSION AND FUTURE WORK

In this chapter we have presented two sets of methods for solving the single-unit process plan generation problem with reconfigurable machines. The first methods are exact ones, comprising two mathematical models. The second method is a heuristic one, comprising heuristics based on mathematical programming. Furthermore, we have proposed methodologies for rapidly obtaining reasonable lower bounds, which are employed to assess the quality of the solutions generated by the other approaches. Our evaluation of the two mathematical modelling approaches indicates that the model with one main decision variable (1MDV) is more effective than the one with two main decision variables (2MDV). The 1MDV model may be employed as a generator of lower bounds or as a benchmark for small-sized instances. However, its applicability is limited to small to medium-sized instances, particularly those with a low number of operations.

As demonstrated by the results of the two approaches, the heuristic technique proved to be an effective method for decision-makers, offering a rapid alternative, and good quality solutions. This suggests that mathematical-based heuristics can be employed to effectively address large-sized instances of the process plan generation problem. As a consequence of the effectiveness of the lower bound approach, the distance from the optimum remained constant across all instances. Nevertheless, in order to reduce this distance from the optimal, further advancements should be made in lower bound computations. Improved lower bounds can be generated in a shorter CPU time and subsequently employed in a Branch-and-Bound algorithm or in conjunction with a heuristic based on Lagrangian relaxation.

The results of this chapter as a whole have demonstrated that the combination of these methods can provide effective ways to solve the Single-Unit process planning problem in a reconfigurable environment. In the next chapter, we will extend this problem to the multi-product case, where a workshop consisting of reconfigurable machines manufactures different components with different requirements.

The proposed approaches can function in an e-Computer-Aided Process Planning (e-CAPP) environment. The Lower bounds can be combined with feature recognition systems to check the feasibility[2] of the production of the proposed product designs on the existing machines. It is possible to extend the lower bounds to include production cost and energy consumption lower bounds.

2 The lower bound only considers precedence constraints, so feasibility in this case is limited to TADs and tool constraints.

5 MULTI-PRODUCT PROCESS PLANNING WITH RMS

This chapter[1] aims at addressing the MPPP in a multi-objective framework, using exact and metaheuristic approaches. The exact approach is based on the NBI scalarization strategy, while the metaheuristics consist of two evolutionary algorithms, specifically, NSGA-II, known for its effectiveness in solving process planning problems (Khan et al. 2022), and MOEA/D. This algorithm is particularly noteworthy for its ability to operate without the computationally expensive calculation of crowding distance, a potential that warrants further investigation.

The rest of this chapter is organised as follows: The first section provides a brief introduction. Following that, we elaborate on the problem under consideration and present the mathematical model. Section 5.3 outlines the three proposed approaches for tackling the problem, with Section 5.4 being dedicated to describing the conducted experiments and interpreting the obtained results. Finally, in the last section, we provide some conclusions.

5.1 INTRODUCTION

The MPPP problem arises in several contexts:

- In environments with high variety and low volume production, where process plans are typically executed as they are.
- In situations where minimizing reconfiguration efforts is a priority (Kant et al. 2020), particularly during transitions between different process plansa factor often overlooked in single-product, single-unit process planning.
- In cases with resource constraints that require products to be processed sequentially (Kant et al. 2020). This is the case when there are limitations on operators, such as in single-manned workstations, or on fixtures, which necessitate the processing of one product at a time.

5.1.1 *Illustrative example*

The core idea of this work can be summarized as follows: when you replicate the optimal process plan of a single unit (optimized independently) for n units, it does

1 This chapter was submitted for publication in an international journal and is currently under review.

not necessarily result in the optimal process plan for manufacturing all n units. This concept is further illustrated through an illustrative example. Consider the scenario where the process plans are generated for manufacturing two units of the same product on 2 RMTs. The processing times for operations, machine changeover time, and configuration changeover time are presented in Tables 5.1, 5.2, and 5.3, respectively.

Operation	M1	M2
OP1	2	2
OP2	4	
OP3		4

Table 5.1: Processing time/possible machines

	M1	M2
M1	0	1
M2	3	0

Table 5.2: Machine changeover time (tu)

M1		C1	C2	
	C1	0	1	
	C2	3	0	
M2		C1	C2	C3
	C1	0	4	4
	C2	4	0	4
	C3	4	4	0

Table 5.3: Configuration changeover time (tu)

A brief examination of Table 5.1 reveals that there are only two possible operations assignments: one in which OP1 is assigned to M1 and another in which it is assigned to M2. These are presented in Tables 5.4 and 5.5.

Operation	OP1	OP2	OP3
Machine	M1	M1	M2
Configuration	C1	C2	C2

Table 5.4: Process Plan A

Operation	OP1	OP2	OP3
Machine	M2	M1	M2
Configuration	C2	C2	C2

Table 5.5: Process Plan B

If we compute the total time of each process, we can see that process A is optimal with 12 tu. Meanwhile, for process B it's 14 tu, since it contains a machine changeover from 2 to 1 with 3 tu.
Even though A is the optimal process plan but when repeated it gives worse results (TT=27) than the process A and B combined (TT=26). The figures 5.1 and 5.2 represent the process plans.

That's due to the fact that when combining A and B we eliminate the change of configuration from 2 to 1 in machine 1. This change is present when process A is

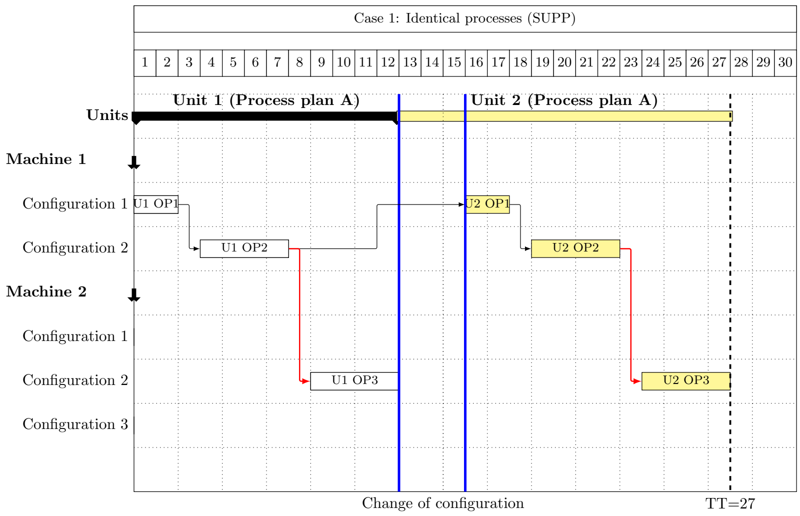


Figure 5.1: Gantt chart of the process plan A repeated twice.

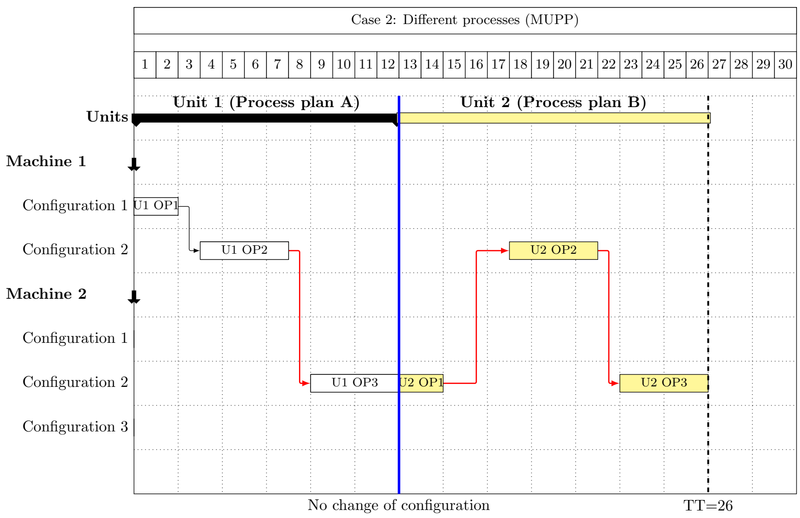


Figure 5.2: Gantt chart of Process plan A and B combined.

repeated, but it's not considered in the mathematical model when the process is optimized (independently) as in the problem treated in chapter 4.

Another example from the literature that can be cited is in the case study proposed in Azab and ElMaraghy, 2007b, the authors generate a process plan for an engine front cover part. Operations 17 and 8 require a particular setup where they add a rotational motion axis to a three-axis RMT. This reconfiguration requires significant time, given the nature of the task. It is clear from the results that the best process plans are those where operations 17 and 8 are in direct succession, as their setup time is large.

If the obtained process plan is used to machine two units, it would be preferable to delay operations that require special setup to the end of the first unit's plan and advance them to the beginning of the second plan. By this, we will set up the machine once every two parts, this is not possible if we had a single process plan repeated.

It should be noted that product sequencing is a scheduling decision, thus MPPP can be considered as a special case of the IPPS problem.

5.2 PROBLEM DESCRIPTION AND FORMULATION

In this section we describe the problem studied, then we present the mathematical model.

5.2.1 *Problem description*

This study tackles MPPP problem in the context of RMTs. The problem involves manufacturing a set of similar products—from the same product family—on a predetermined shop floor. Each product requires a set of operations, which has precedence constraints between them. These constraints, dictate the specific operations that must be completed before others can start. Furthermore, each operation requires a defined set of alternative tools and TADs for execution.

The RMTs are prepositioned in the shop floor, each equipped with a tool magazine containing a selection of usable tools. Each configuration of an RMT provides one or more TADs, enabling the machining of workpieces. The feasibility of an operation on a given machine-configuration depends on the availability of both the required TAD and tool. Consequently, if a machine has at least one of the alternative tools and TADs needed for a specific operation, it can perform that operation.

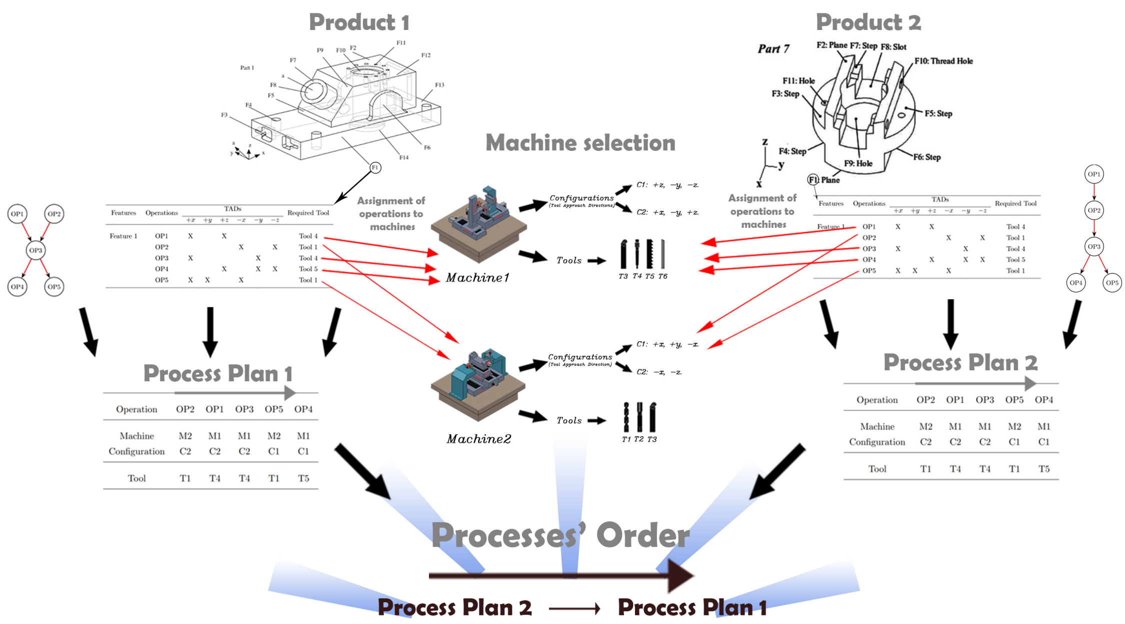


Figure 5.3: Description of the multi-Product process planning problem with the main elements and decisions taken.

As illustrated in Figure 5.3, MPPP problem is divided into two interconnected subproblems: single-product process planning and product sequencing.

The first subproblem, focuses on determining two key aspects: (i) the sequence of operations while respecting precedence constraints, and (ii) the assignment of machine configurations and tools required for each operation.

The second subproblem, product sequencing, involves arranging the process plans of all products in a sequence that optimizes the overall manufacturing efficiency.

The problem has two minimization objectives: total production time and total production cost. Total production time includes the operational processing times on machines, machine changeover times, configuration changeover times, and tool changeover times. Similarly, total production cost accounts for processing costs and machine, configuration, and tool changeover costs. Together, these two objectives aim to balance efficiency and cost-effectiveness in a reconfigurable manufacturing environment.

Table 5.6 provides an illustrative example of a solution to MPPP, presenting both the process plans for individual products and their production sequence (P2 -> P1). The information in the table can be read column by column from left to right. For instance, the first column specifies that operation OP1 of product P2 is executed first on machine M2, using configuration C3 and tool T1. Following this, operation OP4 is performed on machine M1, and the sequence continues accordingly.
The table also highlights changeovers between operations. For example, a machine changeover occurs between the first and second operations, switching from machine M2 to M1. Furthermore, machine M2 undergoes both a configuration change and a tool change between the first and third positions, transitioning from configuration C3 to C2 and from tool T1 to T5. This detailed breakdown illustrates the complexities and dynamic nature of process planning and sequencing in RMSs.

Table 5.6: An example of a MPPP Solution representation.

Product	P2	P2	P2	P2	P1	P1	P1
Operation	OP1	OP4	OP3	OP2	OP2	OP1	OP3
Machine	M2	M1	M2	M1	M2	M2	M1
Configuration	C3	C2	C2	C1	C2	C1	C1
Tool	T1	T1	T5	T3	T1	T2	T1

It is important to highlight that the system does not start from a predefined initial state. As a result, when a machine is used for the first time, no configuration or tool changeover is required. Likewise, no machine changeover is considered for the initial operation of the first product in the sequence.

5.2.2 *Mathematical model*

In this section, we present the problem formulation and the 0-1LP mathematical model.

In addition to the classical indices usually found in SUPP, we introduce the product index, employing a six-dimensional decision variable denoted as $P^{v,u}_{s,j,l,q}$. This variable is set to one if operation u of product v is processed in position s

by machine j, configuration l, and using tool q. The first column of the previous example presented in Table 5.6 can be represented by $P^{2,1}_{1,2,3,1} = 1$.

Notation

To formulate the problem, the following notations are used:

Indices	
v	Index of products
j	Index of machines
l	Index of configurations
u	Index of operations
q	Index of tools
t	Index of TADs
s	Index of positions

Sets	
V	Set of all products
J	Set of all machines
L	Set of all configurations
U	Set of all operations
Q	Set of all tools
TAD	Set of all TADs
S	Set of all positions

Parameters	
N	Number of products to manufacture
M	Number of available machines
T	Number of tools
NC_j	Number of available configurations for machine j
NOP_v	Number of operations of product v
$TNOP$	Total number of all operations of all products
MT_{jq}	equals 1 if machine j can use tool q, 0 otherwise
$CTAD_{jlt}$	equals 1 if machine j in configuration l can machine the workpiece from TAD t, 0 otherwise
$OPTAD_{vut}$	equals 1 if operation u of product v can be executed from TAD t, 0 otherwise
OPT_{vuq}	equals 1 if operation u of product v can be executed with tool q, 0 otherwise
$OPP_{vuu'}$	equals 1 if operation u' has to be processed after operation u of product v
$CCTime_{jll'}$	Configuration change time for machine j from configuration l to l'
$TCTime_{qq'}$	Tool change time from tool q to q'
$MCTime_{jj'}$	Machine change time from machine j to j'
$PrTime_{vujl}$	Processing time for operation u of product v on machine j with configuration l
$CCCost_{jll'}$	Configuration change cost per unit time for machine j from configuration l to l'
$TCCost_{qq'}$	Tool change cost per unit time from tool q to q'
$MCCost_{jj'}$	Machine change cost per unit time from machine j to j'
$PrCost_{vuj}$	Processing cost per unit time for operation u of product v on machine j
B	A big number

Decision variables	
P_{sjlq}^{vu}	equals 1 if operation u of product v is in position s executed by machine j in the configuration l using tool q, 0 otherwise
$Z_{jss'}$	equals 1 if machine j is used in positions s and s', 0 otherwise
$W_{sjj'}$	equals 1 if there is a change from machine j to j' between positions s and $s+1$, 0 otherwise
$X_{sjll'}$	equals 1 if there is a change for machine j from configuration l to l' in position s, 0 otherwise
$Y_{sjqq'}$	equals 1 if there is a change for machine j from tool q to q' in position s, 0 otherwise

Objective functions

f^t and f^c are respectively the total time and total cost, where:

$$\text{Minimize } f^t = \sum_s \sum_v \sum_u \sum_j \sum_l \sum_q P^{vu}_{sjlq} \times PrTime_{vuj} + \sum_s \sum_v \sum_j \sum_{j'} W_{sjj'} \times MCTime_{jj'} + \sum_s \sum_v \sum_j \sum_l \sum_{l'} X_{sjll'} \times CCTime_{jll'} + \sum_s \sum_v \sum_j \sum_q \sum_{q'} Y_{sjqq'} \times TCTime_{qq'} \quad (5.1)$$

$$\text{Minimize } f^c = \sum_s \sum_v \sum_u \sum_j \sum_l \sum_q P^{vu}_{sjlq} \times PrTime_{vujl} \times PrCost_{vuj} + \sum_s \sum_v \sum_j \sum_{j'} W_{sjj'} \times MCTime_{jj'} \times MCCost_{jj'} + \sum_s \sum_v \sum_j \sum_l \sum_{l'} X_{sjll'} \times CCTime_{jll'} \times CCCost_{jll'} + \sum_s \sum_v \sum_j \sum_q \sum_{q'} Y_{sjqq'} \times TCTime_{qq'} \times TCCost_{qq'} \quad (5.2)$$

Constraints

$$Z_{jss'} \geq (1 - \sum_{s''=s+1} \sum_v \sum_u \sum_l \sum_q P^{vu}_{s''jlq}) + (\sum_v \sum_u \sum_l \sum_q P^{vu}_{sjlq} + P^{vu}_{s'jlq}) - 2$$
$$\forall j = 1, ..., M, \quad \forall s = 1, ..., TNOP - 1, \quad \forall s' = s + 1, ..., TNOP \quad (5.3)$$

$$W_{sjj'} \geq \sum_v \sum_u \sum_l \sum_q P^{vu}_{sjlq} + P^{vu}_{s+1j'lq} - 1$$
$$\forall s = 1, ..., TNOP - 1, \quad \forall j, j' = 1, ..., M \quad (5.4)$$

$$X_{sjll'} \geq Z_{jss'} + (\sum_v \sum_u \sum_q P^{vu}_{sjlq} + P^{vu}_{s'jl'q}) - 2$$
$$\forall s, s' = 1, ..., TNOP, \quad \forall j = 1, ..., M, \quad \forall l, l' = 1, ..., NC_j \quad (5.5)$$

$$Y_{sjqq'} \geq Z_{jss'} + (\sum_v \sum_u \sum_l P^{vu}_{sjlq} + P^{vu}_{s'jlq'}) - 2$$
$$\forall s, s' = 1, ..., TNOP, \quad \forall j = 1, ..., M, \quad \forall q, q' = 1, ..., T \quad (5.6)$$

$$OPP_{vuu'} + \sum_j \sum_l \sum_q P^{vu}_{sjlq} + P^{vu'}_{s'jlq} \leq 2$$
$$\forall s = 2, ..., TNOP, \quad \forall s' = 1, ..., s \quad \forall u, u' = 1, ..., NOP_v \quad \forall v = 1, ..., N \quad (5.7)$$

$$P^{vu}_{sjlq} \leq \sum_t CTAD_{jlt} \times OPTAD_{vut}$$
$$\forall s \in S, \forall u \in U, \forall v \in V, \forall l \in L, \forall q \in Q, \forall j \in J \quad (5.8)$$

$$P^{vu}_{sjlq} \leq MT_{jq} \times OPT_{vuq}$$
$$\forall s \in S, \forall v \in V, \forall u \in U, \forall l \in L, \forall q \in Q, \forall j \in J, \forall u \in U \quad (5.9)$$

$$\sum_{j}\sum_{l}\sum_{u}\sum_{q}(P^{vu}_{s-1jlq} - P^{vu}_{sjlq} + 1) \times B \geq \sum_{j}\sum_{l}\sum_{u}\sum_{q}\sum_{s'=1}^{s-2} P^{vu}_{s'jlq} \qquad \forall s = 2, ..., TNOP \tag{5.10}$$

$$\sum_{s}\sum_{j}\sum_{l}\sum_{q} P^{vu}_{sjlq} = 1 \qquad \forall u \in U, \forall v \in V, \tag{5.11}$$

$$\sum_{v}\sum_{u}\sum_{j}\sum_{l}\sum_{q} P^{vu}_{sjlq} = 1 \qquad \forall s \in S, \tag{5.12}$$

$$P^{vu}_{sjlq}, Z_{jss'}, W_{sjj'}, X_{sjll'}, Y_{sjqq'} \in \{0,1\}$$
$$\forall s \in S, \forall v \in V, \forall u \in U, \forall l \in L, \forall q \in Q, \forall j \in J, \forall u \in U \tag{5.13}$$

The first objective function (5.1) minimizes the total production time for all products, encompassing the processing times of operations as well as the changeover times for machines, configurations, and tools. The second objective function (5.2) focuses on minimizing the total production cost, calculated as the sum of time-related costs. This includes the processing costs of operations and the costs associated with machine, configuration, and tool changeovers.

Constraints (5.3) link the use of each machine to its subsequent usage position s' through the variable Z. Constraints (5.4) determine when a machine changeover occurs between two consecutive positions s and $s+1$. Constraints (5.5) and (5.6) identify configuration and tool changeovers, respectively. Constraints (5.7) represent precedence constraintss, ensuring that operations are processed only after their predecessors have been completed. Constraints (5.8) ensure that each machine configuration includes at least one TAD required for the assigned operations. Constraints (5.9) verify that the necessary tool is available on the machine for machining the assigned operations. Constraints (5.10) sequence products in a way that keeps the operations of each product grouped together, so that no operation of a different product could be scheduled inside another product's operations sequence. Constraints (5.11) ensure that every operation of every product is executed exactly once. Constraints (5.12) mandate that only one operation can be performed at a given time of a given product. Constraints (5.13) are integrality constraints.

Additionally, equations (5.14) and (5.15) compute the parameters $TNOP$ and B, respectively.

$$TNOP = \sum_{v} NOP_v \tag{5.14}$$

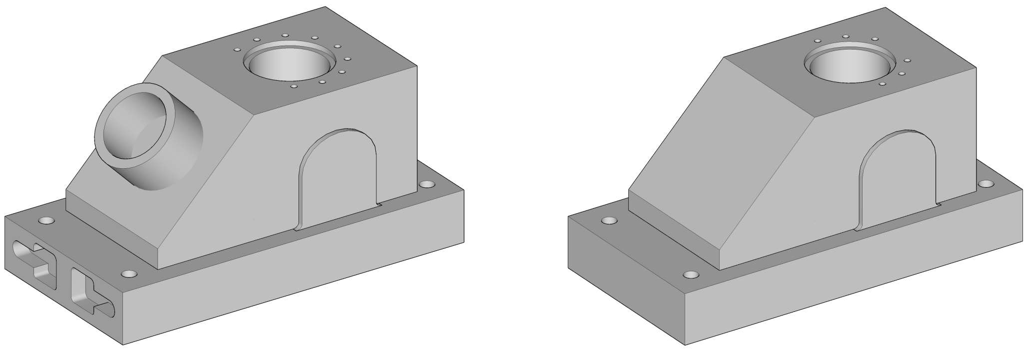
$$B = \max_{v \in \{1,...,N\}} (NOP_v) + 1 \tag{5.15}$$

5.2.3 *Illustrative example*

As a practical illustration, we adapted an example from Youssef and ElMaraghy, 2006. The scenario involves two components: ANC-101 and its variant ANC-90, comprising 20 and 12 operations, respectively. These operations are processed using three reconfigurable machines: a horizontal milling machine (M1) with five configurations, a drilling press (M2) with two configurations, and a boring machine (M3) with a single configuration. Additionally, ten tools are available for machining tasks. Figure 5.4 provides a visual representation of the two parts, while Table 5.7 outlines the capabilities of the three machines.

Machine M1, equipped with five configurations, demonstrates significant flexibility as it can accommodate any of the ten available tools. In the first four configurations, its functionality is restricted to using $+z$ and $-z$ TADs, with the primary distinction among these configurations being the operational speed, determined by the number of spindles.

It is important to note that a machine can only operate in one configuration at a time. Additionally, for an operation to be performed, the selected machine configuration must have access to the required tool and at least one of the necessary TADs. For further details about this example, refer to Youssef and ElMaraghy 2006.



(a): Part ANC-101 (b): Part ANC-90

Figure 5.4: The two example parts ANC-101 and its variant ANC-90 (Youssef and El-Maraghy 2006).

Table 5.7: Candidate machine specifications of the example.

Machines	Configurations	TADs								Tools
		$+x$	$+y$	$+z$	$-x$	$-y$	$-z$	$+a$	$-a$	
M1 Reconfigurable horizontal milling machine	C1			×			×			ALL Tools
	C2			×			×			
	C3			×			×			
	C4			×			×			
	C5	×	×	×	×	×	×	×	×	
M2 Reconfigurable drilling press	C1			×			×			T2,T3, T4,T5
	C2			×			×			
	C3			×			×			
	C4						×			
M3 Reconfigurable boring machine	C1						×			T10

Table 5.8: Results obtained for solving the illustrative example.

N	Selected machines configurations	(Total time; Total cost)	Machine changeover	Configuration changeover	Tool changeover	Total changeovers
1	M1C2 M1C4 M1C5	(1647.5; 898.92)	0	4	17	21
2	M1C2 M1C4 M1C5 M2C4	(1657; 892.43)	1	4	16	21
3	M1C2 M1C4 M1C5 M2C4	(1672; 884.83)	1	4	17	22
4	M1C2 M1C4 M1C5 M2C4	(1674; 722.98)	1	3	16	20
5	M1C2 M1C4 M1C5 M2C4	(1686; 714.39)	1	3	17	21
6	M1C2 M1C4 M1C5 M2C4	(1706; 693.56)	1	4	17	22
7	M1C2 M1C4 M1C5 M2C4 M3C1	(1738; 679.14)	3	4	16	23

Table 5.8 summarizes the seven non-dominated solutions obtained for the illustrative example. For instance, solution 1 achieves a total time of 1647.5 (time units) and a total cost of 898.92 (cost units). In this solution, only machine M1 is utilized, operating in configurations 2, 4, and 5. While this solution minimizes total production time, other solutions prioritize cost efficiency. Since only one machine is used in this case, no machine changeovers occur. However, configuration changes on M1 happen four times, and tools are switched 17 times.

Across all seven solutions, part ANC-101 is consistently processed before part ANC-90. The operation sequences for both parts exhibit minimal variation between solutions, indicating that sequencing decisions remain relatively stable. The trade-off between minimizing time and cost is achieved primarily through variations in machine, configuration, and tool assignments.

The second column of Table 5.8 provides details on the machines and configurations employed. Notably, using machine M1 in configuration C5 is unavoidable, as it is the only setup capable of machining features F11 and F12 of ANC-101. However, this configuration operates at a slower speed. Conversely, configuration C4 offers the fastest processing speed but incurs higher costs due to its reliance on four spindles.

5.3 SOLUTION APPROACHES

This section describes the exact and approximate solution approaches proposed to solve the described MPPP. In the first subsection, we present the Normal Boundary intersection algorithm. Next, we present the coding procedure.

5.3.1 *NBI strategy*

NBI scalarization strategy generates non-dominated solutions for multi-objective problems by solving a series of single-objective subproblems (See Chapter 2 for more detail). To optimize its performance, a dynamic function is added which adjusts parameters in each iteration, taking into account the available CPU time and the time required to solve the previous subproblems. This strategy aims to ensure that NBI method explores uniformly distributed regions of the objective

space while making efficient use of the computational resources.
Unlike other methods, the proposed NBI algorithm does not focus on enumerating the entire Pareto front or defining a fixed number of non-dominated solutions beforehand. Instead, its objective is to maximize the number of non-dominated solutions obtained while maintaining their even distribution in the objective space. Furthermore, the algorithm seeks to allocate the computational budget (CPU time) as equitably as possible across all iterations.
This approach offers two key advantages: it enables efficient exploration of the Pareto front within the constraints of the available CPU time, yielding a more representative distribution of solutions. Additionally, it provides a framework for comparing approximate and exact solution methods under identical computational time constraints.

Figure 5.5 depicts the NBI approach. First, the two anchor points (extremes) are determined by solving the two single objective sub-problems (minimizing total time and minimizing total cost). Then, we formulate the sub-problem and calculate the parameters $\beta_{1,i}$ and $\beta_{2,i}$ using equations (5.16) and (5.17), with the initial values for the first iteration set to $\beta_{1,0} = 0$ and $\beta_{2,0} = 1$. i is the index of iteration (for the first iteration $i = 0$).

$$\beta_{1,i} = \beta_{1,i-1} + \frac{\beta_{2,i-1}}{\lceil \frac{time\_left \times (i+2)}{total\_time - time\_left} \rceil + 1} \tag{5.16}$$

$$\beta_{2,i} = 1 - \beta_{1,i} \tag{5.17}$$

This approach seeks to achieve a uniform search along the span between the two anchor points by leveraging the average time required to solve the subproblems. However, due to the variability in computational times across individual subproblems and the uneven distribution of points along the Pareto front, the resulting solutions may not be perfectly uniform in their distribution.

5.3.1.1 *Execution mode*

The two steps highlighted in blue in Figure 5.5 represent the implementation of the mathematical model within a solver to solve the subproblems. Two distinct methods can be utilized for this purpose:

- Direct Approach (referred to as NBI): This method involves directly solving the mathematical model using the solver.
- Enumerate-and-Solve Approach (we name Normal Boundary Intersection with enumerate and solve strategy (NBI-es)): In this approach, the process is divided into two stages: enumeration of possible products sequences followed by solving and selecting the optimal ones.
  1. Enumerate: This step involves generating all possible $m!$ product sequences, where m represents the total number of products to be scheduled. Each sequence corresponds to a unique ordering of the products.
  2. Solve-and-Select: For each generated sequence, the product sequence variables are fixed in the model by setting $\sum_{jlqu} P^{vu}_{1jlq} = 1$ for the product

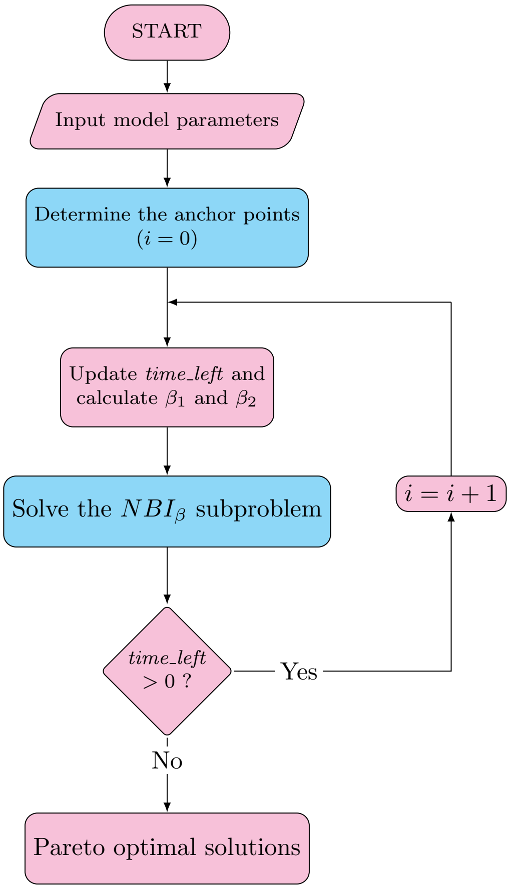


Figure 5.5: The Normal-Boundary Intersection algorithm

v in the first position, a different product v' in the second, and so forth. The solver is then executed to determine the optimal process plans for the specific fixed sequence. This procedure is repeated for all $m!$ sequences. Given the multi-objective nature of the problem, the algorithm compares the resulting solutions and extracts the set of non-dominated solutions, potentially including multiple points for the same sequence (subproblem) on the Pareto front.

This method is feasible only for scenarios involving a small number of products, as the factorial growth of $m!$ with increasing m leads to an exponential increase in computational complexity. Consequently, enumerating and solving for every possible sequence becomes impractical for larger values of m.

5.3.2 *Evolutionary metaheuristics for the MPPP*

In this subsection we present the encoding procedure used to code solutions for the two metaheuristics NSGA-II, MOEA/D as well as the crossover and mutation operators employed.

5.3.2.1 *Chromosome encoding*

The representation of a solution plays a pivotal role in determining both the quality of the final results and the methods used to achieve them. In the context of metaheuristics, this choice is particularly critical, as it directly influences the algorithm's efficiency and effectiveness (Gendreau, Potvin et al. 2010). This challenge becomes even more pronounced in our case, given the vast number of potential solutions and the difficulty of ensuring feasibility while adhering to constraints. For instance, simply swapping two operations can result in an infeasible solution by violating precedence constraints. Moreover, implementing repair mechanisms for infeasible solutions is computationally expensive, and designing genetic operators that consistently produce feasible neighboring solutions is inherently complex (Shabaka and ElMaraghy 2008).

In response to these challenges, Shabaka and ElMaraghy, 2008 introduced a continuous domain variable representation designed to decode into feasible solutions. This encoding method has since been utilized in various studies, including Bensmaine et al., 2013c and Haddou Benderbal et al., 2018. However, the original encoding system does not include product indices, as it was initially developed for SUPP problems. To address this limitation, we extend their approach by incorporating the product index, adding a new row to the matrix representation.
Table 5.9 represents a coded representation of a solution in the form of a $5 \times TNOP$ matrix, where $TNOP$ represents the total number of operations across all products. Each cell in the matrix contains a real number between 0 and 1. Through a decoding process, this coded matrix is transformed into a feasible process plan, similar to the one shown in Table 5.6.
The decoding follows a systematic approach, processing the matrix from left to right, row by row. For each cell, a vector is constructed to enumerate all possible alternatives corresponding to the value could be put in the cell. The value within the cell is then multiplied by the length of this vector, resulting in a number that is rounded up to the nearest integer. This integer serves as an index to select the appropriate alternative from the vector. The process is repeated for every cell in the matrix until a complete and feasible process plan is generated.

Table 5.9: A MPPP Solution representation with P2 processed before P1.

Product	0.71	0.28	0.38	0.89	0.25	0.11	0.46
Operation	0.16	0.92	0.66	0.28	0.47	0.42	0.23
Machine	0.65	0.52	0.17	0.51	0.92	0.10	0.42
Configuration	0.81	0.36	0.56	0.11	0.80	0.26	0.73
Tool	0.03	0.25	0.52	0.31	0.29	0.17	0.62

Figure 5.6 demonstrates the decoding process for the first row of the example provided in Table 5.9. The procedure begins by listing all possible alternatives in a vector. In this case, the first cell can contain either product 1 or product 2. Given two alternatives, the value in the cell (0.71) is multiplied by the total number of alternatives (2). The result is then rounded up to the largest nearest integer, which serves as the index for selecting the corresponding product from the vector. Here,

the rounded value is 2, indicating that product 2 is assigned to the first cell. Since product 2 consists of four operations, the next four cells are automatically filled with the value 2. This ensures that all operations of product 2 are completed before moving on to operations from any other product. This process is repeated

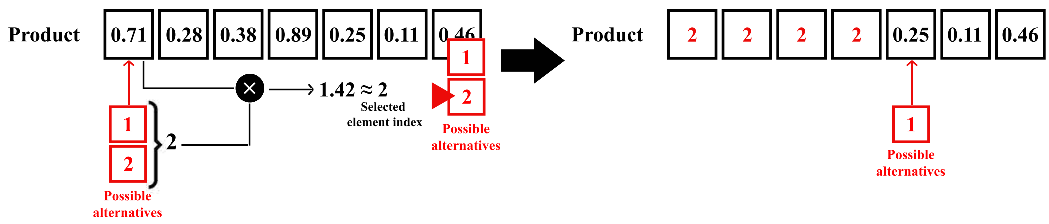


Figure 5.6: Example of the decoding procedure for the vector of products.

across all rows of the matrix to determine the operations, machines, configurations, and tools. While decoding the operations, precedence constraints are taken into account. For machines, the TADs and tool requirements are considered, in addition to the operations listed in the previous row. When selecting configurations, TAD constraints are also considered, along with the machine assignments from the previous row. Finally, when assigning tools, the operations and machines from previous rows are considered, along with the available tools for the selected machine and the tools required for the corresponding operation.

5.3.2.2 *Genetic operators*

Crossover and mutation operators are essential components of evolutionary algorithms, enabling both the exploration and exploitation of the search space. As shown in Figure 5.7, we use a one-point crossover method, where a random index is chosen as the crossover point. The chromosome is then split vertically at this point, and offspring are created by combining segments from both parents. Mutation involves randomly modifying selected genes from the parent to generate a new child, with the number of genes to be altered determined by a pre-defined mutation rate.
For selection, we apply tournament selection. In this method, four chromosomes are randomly chosen and paired for a tournament. They are compared, and the best-performing chromosome is selected as the winner. This process is repeated to select the second parent.

As explained in the encoding procedure in section 5.3.2.1 and the mutation operator in Figure 5.7, the mutation operator may occasionally fail to produce a different child solution from the mutated parent. This occurs when the selected cell for mutation has only one possible alternative (e.g., the last operation in the sequence) or when the mutation does not alter the chosen element from the alternatives vector. To address this issue, a filter function, referred to as *Filter()* in Algorithms 1 and 2, is used to eliminate duplicate solutions from the child population.

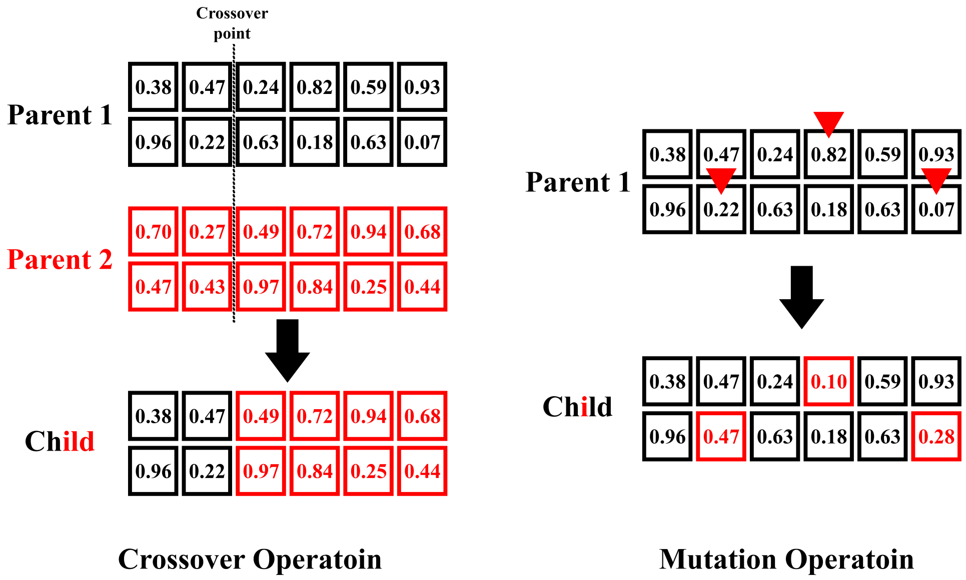


Figure 5.7: Crossover and Mutation operators.

5.4 NUMERICAL EXPERIMENTS

This section begins by describing the generation of experimental tests and the parameter calibration for the two proposed evolutionary metaheuristics. It then evaluates the ability of NBI and NBI-es to generate non-dominated solutions across different CPU time scenarios for small-sized instances. Following this, the performance of MOEA/D and NSGA-II is compared within the same scenarios. Lastly, a comparative analysis between these two metaheuristics is conducted for larger-sized instances.

All algorithms in this section were implemented in Java, with NBI models solved using CPLEX 12.6. The experiments were run on a PC equipped with a 2.9 GHz Intel Pentium processor and 2GB of RAM.

5.4.1 *Experimental setup*

Due to the absence of benchmark instances in the literature for the process plan generation problem with RMS, we conducted experiments using randomly generated instances. Each configuration instance is defined by the total number of products (N), the number of machines (M), and the total number of operations ($TNOP$), represented as (N_TNOP_M).

For each configuration instance, the total number of operations ($TNOP$) is randomly distributed across the N products. For example, in an instance denoted as (2_20_3), there are two products ($N = 2$), 20 operations ($TNOP = 20$), and three machines ($M = 3$). A possible distribution could be 8 operations for product 1 ($NOP_1 = 8$) and 12 operations for product 2 ($NOP_2 = 12$).

To ensure consistency with instances in other works (Haddou Benderbal et al. 2018; Mahmoodjanloo et al. 2020; Shabaka and ElMaraghy 2008; Vahedi-Nouri et al. 2022; Youssef and ElMaraghy 2006), we generated the following parameters in a uniform distribution : $PrTime \in [100, 800]$; $TCTime \in [15, 50]$; $CCTime \in [10, 50]$; $MCTime \in [5, 64]$; $PrCost \in [0.1, 0.8]$; $TCCost \in [0.01, 0.04]$; $CCCost \in [0.01, 0.04]$; $MCCost \in [0.05, 0.1]$; $NC_j \in [1, 5]$; $T = 10$; Moreover, we set the average machine-configuration combinations capable of processing an operation as 20%.

5.4.2 *Parameter tuning*

The performance of metaheuristics is highly dependent on the proper configuration of their parameters. Even a well-crafted metaheuristic can underperform if its parameters are not correctly set. Therefore, in this subsection, we focus on calibrating the parameters of the two evolutionary metaheuristics.
To begin, we identified the parameters that need calibration for each metaheuristic, as shown in Table 5.10. Both metaheuristics share common parameters, such as population size, crossover probability, and mutation ratio. However, MOEA/D introduces an additional key parameter, T, which represents the number of solutions in a neighbourhood group from which parents are selected for crossover to produce offspring. Each of these parameters is has three distinct levels, as outlined in Table 5.10.

Table 5.10: Table of parameters and their levels.

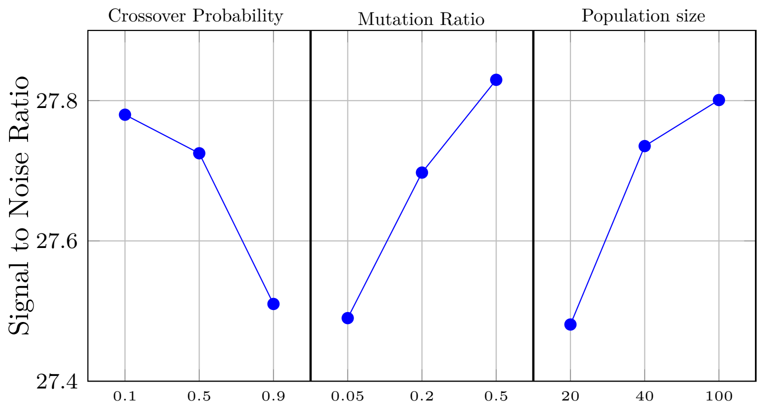
Criteria	Crossover Probability	Mutation Ratio	Population size	T
Level 1	0.9	0.5	100	20
Level 2	0.5	0.2	40	10
Level 3	0.1	0.05	20	5

For NSGA-II, the total number of possible parameter combinations is 27 (3^3), while for MOEA/D, it is 81. Since testing all these combinations would be computationally prohibitive, we employ the Taguchi method (Phadke 1995) to efficiently explore the parameter space in a feasible time. An orthogonal array L_9 is utilized to select 9 distinct parameter combinations for each metaheuristic.
These experiments were conducted on three generated instances (2_10_3, 2_15_2, and 3_15_5), and each test was repeated 10 times to ensure reliability. To assess the effect of parameter variations on HV, we calculate the Signal-to-Noise Ratio (SNR) (Phadke 1995), defined as follows, where y_i represents the HV obtained from experiment i and n is the number of experiments:

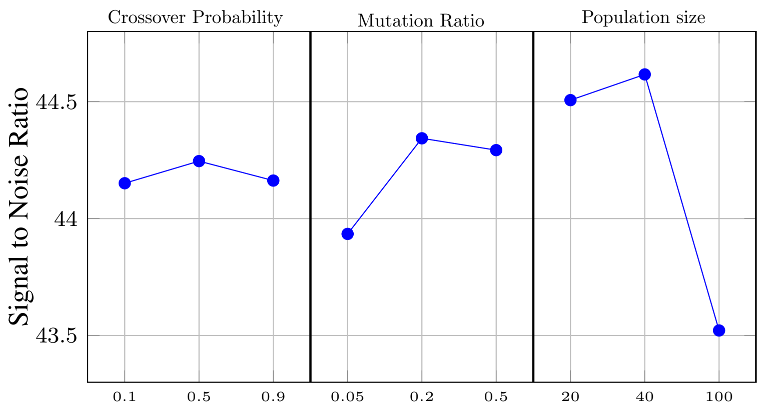
$$\text{SNR} = -10 \times \log \left(\frac{\sum_{i=1}^{n} \left(\frac{1}{y_i^2} \right)}{n} \right)$$

The effect of parameter variation on SNR for the NSGA-II tests is shown in Figure 5.8. The parameter settings that result in higher SNR values are considered the most effective. As illustrated in Figure 5.8, a crossover probability of 0.5 yielded the highest SNR for the second and third instances, while ranking second for the first instance. As a result, we selected this value (0.5) for the crossover probability. Similarly, a mutation ratio of 0.5 also performed well, and a population size of 40 exhibited strong SNR results.

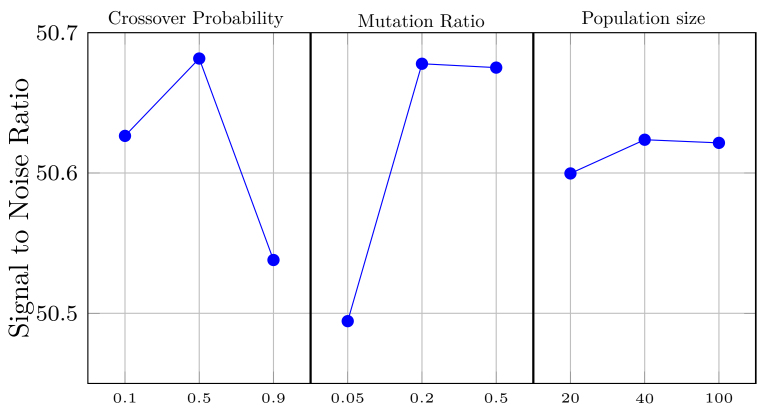
The results from the MOEA/D tests are shown in Figure 5.9. The optimal parameter values for MOEA/D were found to be a population size of 20, a crossover probability of 0.1, a mutation ratio of 0.2, and a value of 5 for T, which yielded the best performance.



(a): Instance 2_10_3

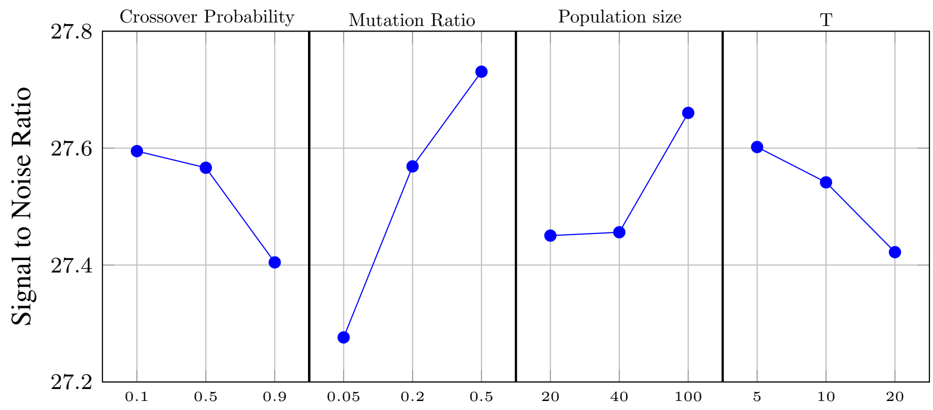


(b): Instance 2_15_2

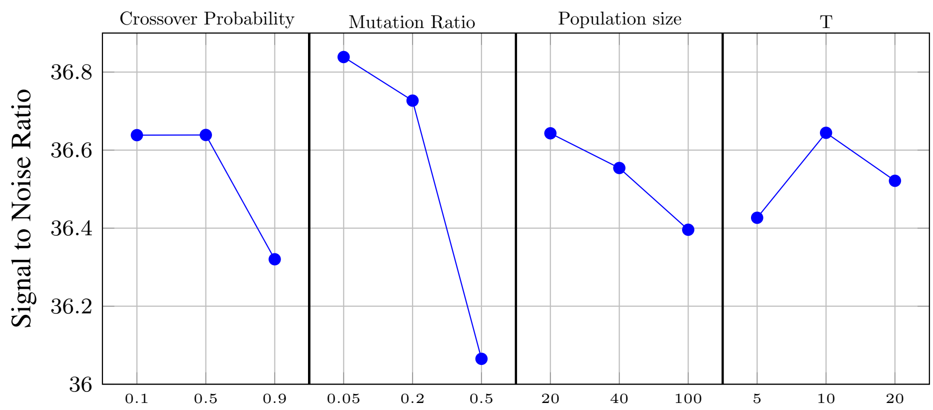


(c): Instance 3_15_5

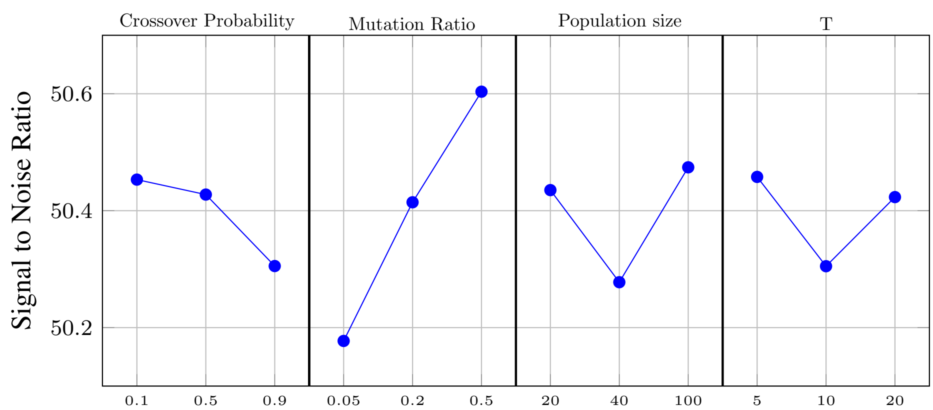
Figure 5.8: Plot of SNR variation as function of different parameters of NSGA-II.



(a): Instance 2_10_3



(b): Instance 2_15_2



(c): Instance 3_15_5

Figure 5.9: Plot of SNR variation as function of different parameters of MOEA/D.

5.4.3 *Comparing the solution approaches for small size instances with different CPU times*

In this subsection, we assess the performance of the two evolutionary metaheuristics using their optimal parameter values, as well as the NBI approaches, under varying computational times for small-sized instances.
Through initial tests and existing research (Khezri et al. 2021; Touzout and Benyoucef 2019a), we observed that no standard method for setting CPU time exists, and applying a fixed CPU limit would not be suitable, as the minimum required computational time varies significantly across instances. Additionally, the problem's complexity is highly influenced by parameters such as $TNOP$ and M.
To address this, we propose a formula to calculate CPU time that ensures a fair comparison across different instance sizes and solution methods. The computational time is given by the formula: $TNOP \times M \times C_{cpu}$ (in seconds), where C_{cpu} is a constant integer. By adjusting the value of C_{cpu}, we generate various scenarios with different computational budgets.

We conducted tests on small instances across five different scenarios for the values of C_{cpu}: 1, 2, 3, 6, and 12. The instance configurations tested (N'_TNOP_M) include 2_8_2, 2_10_2, 2_10_3, and 2_15_2. For each configuration, we generated 5 random instances.
In contrast to the deterministic nature of the NBI approaches, the metaheuristics exhibit stochastic behaviour. Therefore, we performed 10 runs for each instance and each metaheuristic, and the average result was recorded.
To assess the quality of the results, we combined all the obtained fronts into one reference front, representing the Pareto front approximation. After calculating the HV for the reference front and each obtained front, we computed the gaps between the HVs (HV_i) and the reference front HV (*Ref.* HV) using the following formula:

$$\text{Gap for solution } i = \frac{\text{Ref. } HV - HV_i}{\text{Ref. } HV} \times 100$$

In addition to HV gaps, the results were compared on two other metrics: SM and the number of points in the front (NS). See chapter 2 for more detail about performance metrics.

When the obtained front consists of only one non-dominated solution, calculating SM becomes problematic, as the formula yields an undefined value ($SM = \infty$). To resolve this, we assign a fixed value of 2 $SM = 2$ in such cases. However, another issue arises in instances where the two objectives are in harmony, resulting in a single Pareto-optimal solution. This situation affects the average SM values, particularly for methods like NBI, which often identify the single Pareto-optimal solution. To address this, instances with only one Pareto-optimal solution are excluded from the average SM calculation. The results are presented in Figure 5.12. It is worth mentioning that only 4 out of the 20 instances in the small-size configuration category exhibit this characteristic.

Table 5.11 summarizes the average gap to *Ref.* HV across the five instances for each configuration, considering various C_{cpu} values. Since the metaheuristics were run 10 times per instance, the table also reports the average Coefficient of variation

(CV) across the five instances. The best (i.e., smallest) gap for each configuration and each C_{cpu} value is highlighted in bold.

Table 5.11: HV results for the four approaches (HV gap percentages to the reference front).

Configuration	C_{cpu}	NBI Gap	NBI-es Gap	NSGA-II		MOEA/D	
				Gap	CV	Gap	CV
2_8_2	1	**2.46**	7.80	15.43	0.15	14.71	0.18
	2	**1.73**	2.16	8.47	0.12	15.37	0.22
	3	1.71	**1.42**	7.64	0.08	14.52	0.17
	6	0.11	**0.04**	4.70	0.06	11.54	0.14
	12	**0.02**	0.03	3.98	0.06	10.47	0.16
2_10_2	1	34.65	22.99	14.82	0.15	**13.11**	0.11
	2	11.06	**0.69**	7.81	0.08	9.93	0.08
	3	1.83	**0.14**	9.09	0.09	9.76	0.09
	6	0.73	**0.10**	7.74	0.08	8.31	0.07
	12	0.05	**0.04**	5.94	0.07	6.64	0.08
2_10_3	1	19.41	19.33	**1.87**	0.03	5.83	0.10
	2	18.26	19.34	**2.52**	0.06	6.10	0.09
	3	0.46	**0.17**	2.04	0.06	6.56	0.10
	6	**0.03**	0.17	1.96	0.06	3.14	0.07
	12	**0.00**	0.15	0.18	0.00	1.67	0.04
2_15_2	1	73.16	63.47	31.63	0.15	**16.07**	0.07
	2	67.54	60.19	25.60	0.11	**13.57**	0.08
	3	62.78	58.70	23.15	0.14	**13.01**	0.07
	6	62.21	58.52	20.72	0.13	**11.29**	0.07
	12	61.74	57.64	16.26	0.08	**8.97**	0.07

When comparing the two exact solution approaches, NBI and NBI-es, it is evident that NBI-es generally outperforms NBI, indicating that the Enumerate-and-Solve strategy enhances solver performance. However, it is worth noting that NBI shows better results for certain configurations when CPU time is limited ($C_{cpu} = 1$ or 2).

For the 2_10_3 instances with small CPU time ($C_{cpu} = 1, 2$), NSGA-II achieves the best performance among all solution methods. On the other hand, for the largest instances (2_15_2), MOEA/D significantly surpasses all other approaches, regardless of the allocated CPU time. For these large instances, the average gaps to the reference HV for NBI and NBI-es are 61.74% and 57.64%, respectively, even with the highest CPU time ($C_{cpu} = 12$) these gaps are notably larger than those observed for the metaheuristics under the smallest CPU time ($C_{cpu} = 1$), where NSGA-II achieves a gap of 31.63% and MOEA/D achieves an impressive 16.07%.

Both NSGA-II and MOEA/D demonstrate improved stability, as indicated by lower CVs, with increased computational time. Notably, NSGA-II shows a steeper reduction in CV, suggesting more consistent convergence. This trend is reinforced by its superior performance in the three smallest configurations, achieving an average gap of 0.18% with a CV of 0 for 2_10_3, compared to MOEA/D's 1.67% gap and 0.04 CV. Interestingly, this trend reverses for the largest configuration (2_15_2), where MOEA/D handles the complexity more effectively, achieving a 16.07% average gap at the minimum C_{cpu}, outperforming NSGA-II even at the maximum

C_{cpu} (16.26% gap). This indicates that MOEA/D may be better suited for larger configurations, a hypothesis that will be further explored in the next subsection.

The 2_10_3 configuration presents a unique case where both metaheuristics achieve near-optimal Pareto fronts at $C_{cpu} = 12$. This contrasts with the results for the other smaller configurations, where NBI approaches generally outperform the metaheuristics. The resulting gaps for NBI, NBI-es, NSGA-II, and MOEA/D are 0%, 0.15%, 0.18%, and 1.67%, respectively.
Examining individual instances within the 2_10_3 configuration reveals that 2 out of 5 instances exhibit no conflict between objectives. For these instances, metaheuristics effectively close the gap to the true Pareto front HV at large CPU times, as no additional points are discoverable by NBI approaches beyond the single optimal solution.

For instances with conflicting objectives, metaheuristics tend to perform better. However, in non-conflicting cases, both NSGA-II and MOEA/D struggle to deliver high-quality solutions at lower C_{cpu} values, with notable improvements only observed at $C_{cpu} = 12$. NSGA-II's performance, for example, stagnates or even deteriorates for C_{cpu} values between 1 and 6. Meanwhile, MOEA/D shows worse HV gaps than NSGA-II and experiences a decline in performance between $C_{cpu} = 1$ and $C_{cpu} = 3$.
These findings highlight the importance of considering the level of conflict between objectives when selecting optimization algorithms. NSGA-II's superior handling of conflicting objectives suggests a stronger exploitation capability compared to MOEA/D.

Across all methods, increased computational time leads to reduced average gaps (greater closeness to Pareto fronts) and lower CVs for metaheuristics. This is intuitive, as more processing time allows for more refined searches and greater result consistency. However, there remains a trade-off between exact and approximate methods. Exact approaches perform poorly with limited computational time, while metaheuristics often outperform exact methods in such scenarios but cannot guarantee optimal solutions even with extended computational time. This highlights the classic challenge of balancing solution quality against computational time, as visually depicted in Figure 5.10.

Figures 5.10, 5.11, and 5.12 presents the average results for the three smallest configurations, since for the instances of configuration 2_15_2, the NBI approaches couldn't solve more than one subproblem within the computational time limits, indicating a significant increase in problem complexity for this configuration. Figure 5.10 presents the average HV gap percentage to the reference front per different C_{cpu} values, whereas 5.11 and 5.12 present SM and NS averages, respectively.

Figure 5.10 shows that the two NBI approaches tend to perform worse than the two metaheuristics for low computational budgets ($C_{cpu} = 1$). However, it surpasses them for larger budgets, achieving almost 0% gap for C_{cpu} larger than 6. For the two metaheuristics a similar behaviour can be observed, with NSGA-II consistently outperforming MOEA/D. At the largest computational time ($C_{cpu} = 12$) NSGA-II achieves an average gap of 4.79%, while MOEA/D reaches 7.65%.

Interestingly, NBI and NBI-es experience a sharper decline in the HV gap between $C_{cpu} = 1$ and $C_{cpu} = 3$, dropping from 18.83% to 1.31% for NBI and

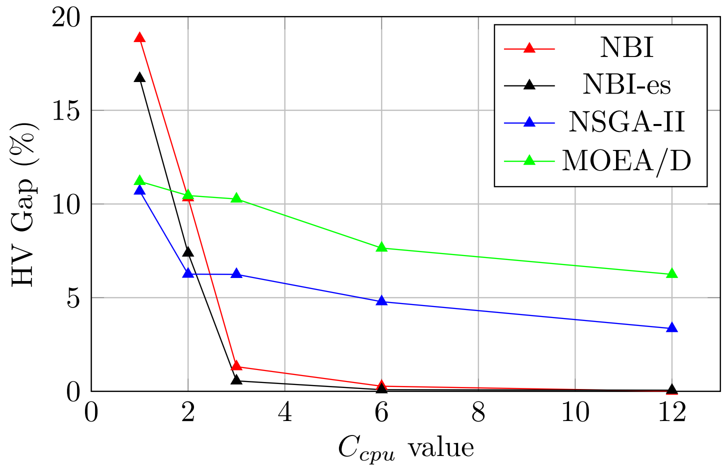


Figure 5.10: Comparing average HV gap of NSGA-II, MOEA/D, NBI, and NBI-es for the three smallest configurations 2_8_2, 2_10_2, and 2_10_3 (N_TNOP_M).

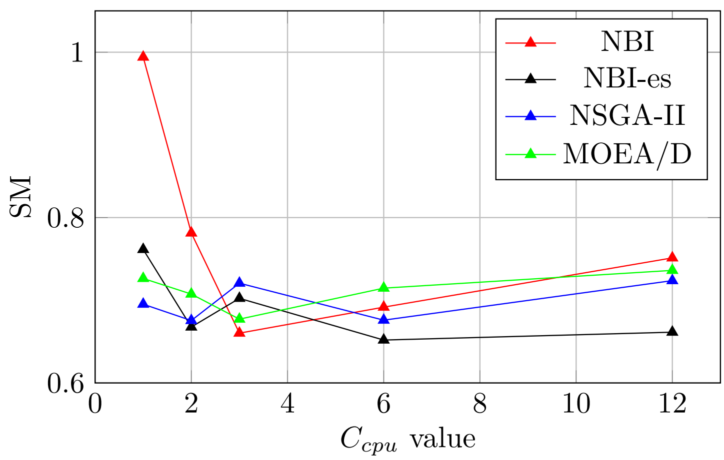


Figure 5.11: Comparing average SM of NSGA-II, MOEA/D, NBI, and NBI-es for the three smallest configurations 2_8_2, 2_10_2, and 2_10_3 (N_TNOP_M).

from 16.70% to 0.56% for NBI-es. This is due to the NBI method procedure with initial generation of the two extreme (anchor) points that significantly contribute to the HV value. Subsequent points are iteratively generated, prioritizing those furthest from existing points. This suggests that the prioritization results in points generated in later iterations being closer to existing points, thus contributing less to the overall HV. This is evident in the minimal decrease in the HV gasp for NBI and NBI-es when C_{cpu} doubles from 6 to 12.

Figure 5.10 also reveals that NBI-es consistently outperforms NBI in terms of HV for computational budgets of 1, 2, and 3. This superiority is further evidenced by Figure 5.12, which demonstrates that NBI-es generates a larger number of non-dominated solutions than NBI across all C_{cpu} values. The better performance of NBI-es can be attributed to its ability to identify more than one non-dominated solution within each subproblem. This suggests that NBI-es is more effective in exploring the solution space compared to NBI.

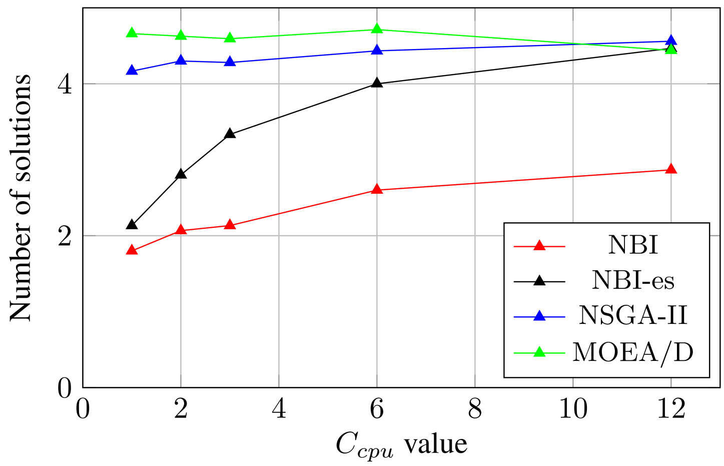


Figure 5.12: Comparing average NS of NSGA-II, MOEA/D, NBI, and NBI-es for the three smallest configurations 2_8_2, 2_10_2, and 2_10_3 (N_TNOP_M).

Figure 5.11 reveals that the NBI approaches, particularly NBI-es, generate Pareto fronts with a more evenly distributed spread of solutions compared to the meta-heuristics. This is most evident for the highest computational budget ($C_{cpu} = 12$) where NBI-es achieves the lowest SM value of 0.66.

As mentioned earlier, for C_{cpu} of 6 and 12, the two NBI methods has almost 0 HV gaps. This is despite the increase in the number of non-dominated solutions for these C_{cpu} values, especially for NBI-es, as evident in Figure 5.12. This suggests that the additional solutions discovered by NBI-es are located in regions of the Pareto front with minimal impact on the overall HV, for example near the already discovered non-dominated points. However, Figure 5.11 reveals a different trend. The SM remains relatively stable for NBI-es between C_{cpu} 6 and 12, suggesting that the additional solutions are distributed uniformly in the existing Pareto front, with minimal impact on the overall solution spread.

This phenomenon is likely due to the staircase structure of the Pareto front, this is further confirmed by the Pareto front described in Figure 5.13, where the additional points found for $C_{cpu} \geq 3$ align with the HV space (see Chapter 2, Figure **??**) and thus contributing marginally to the overall HV.

Figure 5.13 depicts the Pareto fronts obtained by the NBI algorithm for various C_{cpu} values for instance 5 of configuration 2_8_2, and the best overall solutions found by the metaheuristics. This figure emphasizes the importance of having a good representation of the true pareto front for solution evaluation. As the NBI algorithm progressively reveals the true Pareto front, the quality of obtained solutions can be better evaluated, allowing for easier identification of poor solutions or fronts.

Figure 5.13 illustrates how the NBI method generates distinct Pareto fronts as the computational budget (C_{cpu}) increases. A larger computational budget enables the identification of more non-dominated solutions with improved distribution.
In subfigure 5.13a ($C_{cpu} = 1$), the NBI method discovers the two anchor points and one additional intermediate solution. As the computational time increases (Subfigures 5.13b and 5.13c), a fourth Pareto point emerges. Notably, while the

three solutions from 5.13a reappear in 5.13b and 5.13c, the fourth point differs between these two subfigures. This demonstrates the NBI methods ability to adapt to varying computational budgets by exploring different regions of the search space, enhancing solution diversity.
With $C_{cpu} = 6$ (Subfigure 5.13d), the NBI method identifies 8 non-dominated solutions. Finally, in Subfigure 5.13e, the most diverse Pareto front is achieved with 13 non-dominated solutions generated under the maximum computational time ($C_{cpu} = 12$).

The variability in CPU time required for solving each iteration is a notable challenge that the NBI method effectively manages. For instance, generating the anchor points in this example involves solving two optimization problems: minimizing total production time (solved in 9 seconds) and minimizing total production cost (solved in 5 seconds). Despite the unpredictable nature of the solver's computational time across iterations, the NBI approach consistently achieves relatively uniform spacing among the solutions across all C_{cpu} scenarios, ensuring balanced Pareto fronts.

Figure 5.13 highlights the capability of both NSGA-II and MOEA/D to generate solutions close to the true Pareto front. The red and blue points represent the best solutions achieved by each metaheuristic, based on HV metric averaged over 10 runs. HV values for NSGA-II and MOEA/D are 11,449 and 11,206, respectively. In comparison, NBI solution in subfigure 5.13a achieves an HV of 8,715. However, NBI solution in subfigure 5.13e, obtained with the maximum computational budget ($C_{cpu} = 12$), surpasses the metaheuristics, achieving an HV of 11,644.
The staircase structure of the Pareto front provides further insights into the decision-making process. Solutions on the same plateau share identical operation-to-machine assignments, indicating that variations in product and operation sequences, as well as tool and configuration selections, minimally impact total cost. In contrast, altering operation-to-machine assignments results in transitions between plateaus, significantly influencing total cost.
This observation suggests that, within a hierarchical planning framework, prioritizing operation-to-machine assignments could effectively minimize total costs. Such an approach aligns with the heuristic framework proposed in Mechaacha et al., 2024 and presents a promising area for further exploration.

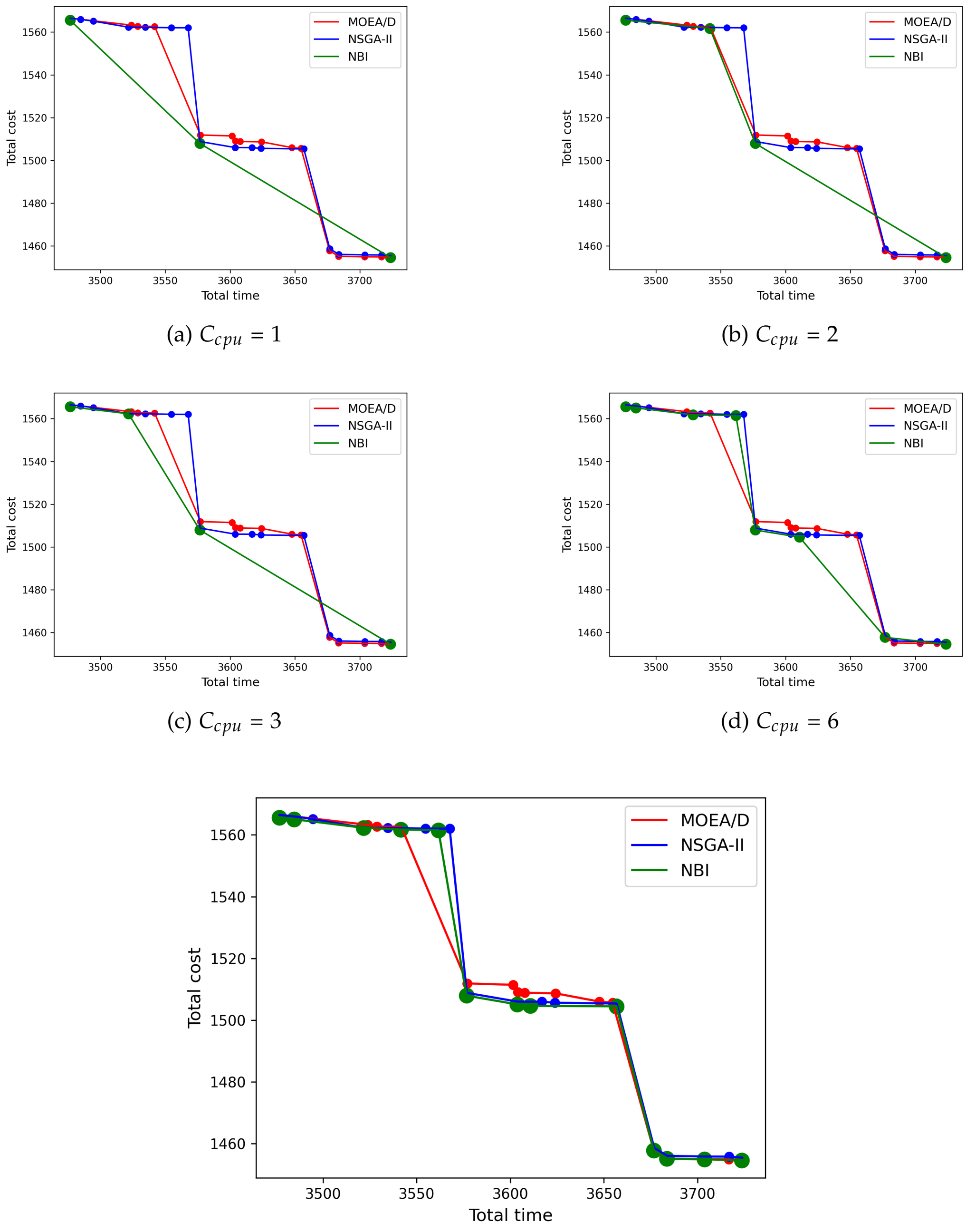
From the results in this section, three key conclusions emerge:

1. NBI-based methods exhibit strong performance on smaller instances, with the enumerate-and-solve strategy further enhancing their effectiveness.

2. When larger computational budgets are available ($C_{cpu} \geq 3$), NBI and NBI-es achieve higher HVs than the metaheuristics for the smallest instances.

3. The metaheuristics demonstrate good performance, achieving HV gaps of approximately 4% and 7% from the reference front for the three smallest configurations, validating their ability to approach exact solution quality.

Given that NBI is not feasible for medium or large problem instances, the next subsection will exclusively evaluate the performance of NSGA-II and MOEA/D on these larger problem sizes.



(a) $C_{cpu} = 1$

(b) $C_{cpu} = 2$

(c) $C_{cpu} = 3$

(d) $C_{cpu} = 6$

(e) $C_{cpu} = 12$

Figure 5.13: The non-dominated solutions generated by the three approaches under different computational budgets for the instance 5 of configuration 2_8_2 (N_TNOP_M).

5.4.4 *Comparing two metaheuristics*

The behaviour and performance of MOEAs are predominantly explored through experimentation, given the complexities of theoretical analysis (H. Li and Zhang 2008). This subsection presents the results of testing NSGA-II and MOEA/D on medium to large-sized instances. Based on the performances shown in Figures 5.10, 5.11, and 5.12, the computational budget (C_{cpu}) is set to 2. This choice strikes a balance between solution quality and computational efficiency, as increasing C_{cpu} beyond 2 offers negligible improvements in solution quality while significantly increasing computation time.

In addition to evaluating performance using HV, SM, and NS, the C-metric is introduced as an additional metric for comprehensive assessment.
The evaluation involved 5 randomly generated instances for each of the 10 configurations. Each instance was run 10 times for both metaheuristics. Table 5.12 presents the average values of the four metrics (HV, C-metric, SM, and NS) across all runs for each instance. Furthermore, it includes the average values for each configuration across all instances.
To determine the statistical significance of the differences observed between the two metaheuristics, the Wilcoxon signed-rank test (Wilcoxon 1947) was conducted at a significance level of 0.05. Instances where one metaheuristic achieved a statistically significant difference are highlighted in bold in the table. Additionally, the number of instances within a configuration where a specific metaheuristic demonstrated significant superiority is indicated in parentheses next to the configurations average value in the "AVG" row.

Table 5.12 highlights MOEA/D's superior performance over NSGA-II across most metrics and configurations.

- HV: MOEA/D consistently achieves higher average HV values across all configurations. This superiority is statistically significant in 45 out of 50 instances. For the largest configuration (10_100_20), MOEA/Ds average HV is 15,913—more than double NSGA-IIs 6,823—indicating that MOEA/D's Pareto front approximations span a substantially larger area relative to NSGA-II points.

- C-metric: MOEA/D generally outperforms NSGA-II in terms of dominance relationships, as reflected by its higher C-metric values. This advantage is statistically significant in 46 out of 50 instances, with no significant difference observed in the remaining four.

- SM: While the differences in SM are less pronounced compared to HV and C-metric, MOEA/D still achieves better average SM values in all configurations except one (4_20_5). Statistically significant differences favour MOEA/D in 8 instances, while NSGA-II demonstrates a significant difference in just 1 instance. This indicates a generally better distribution of solutions within MOEA/Ds Pareto fronts, however the difference is not as pronounced as it was in previous metrics.

- NS: The trend for NS differs from the other metrics. MOEA/D generates more non-dominated solutions for smaller instances, whereas NSGA-II outperforms for larger instances. This divergence could stem from the increasing quality gap between the two metaheuristics for larger instances. As this gap widens, MOEA/D struggles to maintain both high quality and diversity, potentially reducing its NS count. In contrast, NSGA-II may more easily produce a larger number of non-dominated solutions, albeit of lower overall quality, in these scenarios.

Table 5.12: Comparing the performance of the two metaheuristics NSGA-II and MOEA/D for different metrics.

Confi-gura-tion	Inst-ance	HV		C-metric		SM		NS	
		NSGA-II	MOEA/D	C(NSG,MOE)	C(MOE,NSG)	NSGA-II	MOEA/D	NSGA-II	MOEA/D
3_15_5	1	116.21	**151.95**	0.03	**0.69**	0.74	0.63	4.00	**6.40**
	2	157.77	157.49	0.42	0.40	0.69	0.70	23.90	24.10
	3	306.96	302.95	0.48	0.42	0.66	0.63	9.10	10.20
	4	470.51	483.23	0.16	**0.60**	0.57	0.59	12.70	14.00
	5	167.99	172.42	0.37	0.45	0.61	0.67	9.60	15.90
	AVG	243.89(0)	**253.61(1)**	0.29(0)	**0.51(2)**	0.65(0)	0.64(0)	11.86(0)	**14.12(1)**
3_20_3	1	372.69	**457.32**	0.05	**0.90**	0.76	0.69	5.90	7.30
	2	92.28	**114.78**	0.05	**0.87**	0.56	0.73	5.70	**8.40**
	3	186.59	**423.72**	0.02	**0.95**	0.69	0.50	4.70	**8.70**
	4	218.73	**233.77**	0.11	**0.78**	0.67	0.51	9.30	**13.20**
	5	346.15	**475.60**	0.03	**0.91**	0.65	0.56	9.10	12.90
	AVG	243.29(0)	**341.04(5)**	0.05(0)	**0.88(5)**	0.67(0)	0.60(0)	6.94(0)	**10.10(3)**
4_20_5	1	600.57	**657.41**	0.06	**0.87**	0.51	0.52	7.50	11.60
	2	153.07	**167.48**	0.03	**0.82**	0.64	0.57	6.60	8.80
	3	244.37	**382.88**	0.02	**0.93**	0.46	0.61	5.60	**11.00**
	4	87.06	**168.04**	0.00	**0.98**	0.55	0.54	4.60	**7.00**
	5	126.32	**198.26**	0.03	**0.89**	**0.54**	0.67	8.60	**16.90**
	AVG	242.28(0)	**314.81(5)**	0.03(0)	**0.90(5)**	**0.54(1)**	0.58(0)	6.58(0)	**11.06(3)**
4_25_3	1	202.42	**562.74**	0.00	**0.97**	0.57	0.45	5.70	6.30
	2	642.36	**731.16**	0.13	**0.74**	0.61	0.60	13.70	21.60
	3	497.43	**735.44**	0.02	**0.97**	0.57	0.52	8.80	10.90
	4	281.78	**832.49**	0.00	**1.00**	0.51	0.44	5.30	6.50
	5	268.26	**462.07**	0.01	**0.85**	0.59	0.59	7.20	**12.70**
	AVG	378.45(0)	**664.78(5)**	0.03(0)	**0.91(5)**	0.57(0)	0.52(0)	8.14(0)	**11.60(1)**
5_25_5	1	575.90	**1056.22**	0.00	**1.00**	0.53	0.56	11.00	8.10
	2	731.52	**1091.33**	0.01	**0.93**	0.51	0.56	7.00	**12.60**
	3	539.08	**713.85**	0.04	**0.89**	0.57	0.56	14.70	**24.10**
	4	460.04	**826.71**	0.00	**1.00**	0.65	0.52	8.60	11.60
	5	580.55	**929.70**	0.02	**0.92**	0.66	0.58	11.20	**16.90**
	AVG	577.42(0)	**923.56(5)**	0.01(0)	**0.95(5)**	0.59(0)	0.56(0)	10.50(0)	**14.66(3)**
5_50_5	1	3133.21	**4312.63**	0.11	**0.63**	0.50	0.50	10.40	7.70
	2	3768.06	**7187.13**	0.07	**0.89**	0.64	**0.41**	6.10	5.20
	3	2824.73	**4614.75**	0.16	**0.62**	0.50	0.43	6.00	6.50
	4	3116.53	**5557.79**	0.12	**0.82**	0.68	**0.44**	9.20	4.70
	5	3962.53	**6236.05**	0.14	**0.73**	0.67	**0.35**	8.20	5.10
	AVG	3361.01(0)	**5581.67(5)**	0.12(0)	**0.74(5)**	0.60(0)	**0.43(3)**	7.98(0)	5.84(0)
5_30_10	1	1871.69	**4103.78**	0.02	**0.96**	0.59	0.48	9.10	9.00
	2	3018.08	**4805.60**	0.01	**0.95**	0.54	0.52	12.00	11.30
	3	708.29	**1788.87**	0.01	**0.83**	0.51	0.49	8.40	9.50
	4	1660.07	**2419.13**	0.01	**0.90**	0.56	0.51	15.10	11.30
	5	1934.23	**3864.12**	0.00	**1.00**	0.54	0.44	8.90	8.50
	AVG	1838.47(0)	**3396.30(5)**	0.01(0)	**0.93(5)**	0.55(0)	0.49(0)	10.70(0)	9.92(0)
10_50_10	1	1129.80	**4682.76**	0.00	**1.00**	0.69	**0.41**	10.10	6.30
	2	2653.88	**6843.16**	0.01	**0.91**	0.49	0.45	11.10	6.70
	3	1847.56	**5498.60**	0.02	**0.91**	0.63	**0.45**	9.20	5.80
	4	1933.32	**5735.10**	0.00	**1.00**	0.47	0.46	8.00	4.80
	5	3601.35	**6518.65**	0.11	**0.72**	0.65	**0.48**	15.40	8.60
	AVG	2233.18(0)	**5855.65(5)**	0.03(0)	**0.91(5)**	0.59(0)	**0.45(3)**	10.76(0)	6.44(0)
10_75_10	1	1988.48	**6341.96**	0.00	**0.98**	0.58	0.57	5.30	3.90
	2	3697.34	**8278.20**	0.06	**0.82**	0.53	0.52	**10.80**	5.10
	3	8244.42	**14911.64**	0.11	**0.70**	0.53	0.45	**11.40**	7.00
	4	3912.10	**9256.89**	0.05	**0.83**	0.63	0.44	7.30	5.60
	5	2389.80	**6040.72**	0.10	**0.82**	0.68	**0.36**	4.70	4.60
	AVG	4046.43(0)	**8965.88(5)**	0.06(0)	**0.83(5)**	0.59(0)	**0.47(1)**	**7.90(2)**	5.24(0)
10_100_20	1	11711.30	**27542.27**	0.03	**0.85**	0.42	0.42	6.90	6.20
	2	5074.48	**15313.34**	0.02	**0.89**	0.57	0.47	8.30	7.10
	3	4993.16	**10735.35**	0.11	**0.66**	0.55	0.40	8.80	5.10
	4	5411.58	**16727.14**	0.02	**0.90**	0.52	0.43	9.30	7.40
	5	6928.70	9251.29	0.25	0.48	0.55	**0.31**	9.30	4.90
	AVG	6823.85(0)	**15913.88(4)**	0.09(0)	**0.76(4)**	0.52(0)	**0.41(1)**	8.52(0)	6.14(0)

NSG=NSGA-II; MOE=MOEA/D;

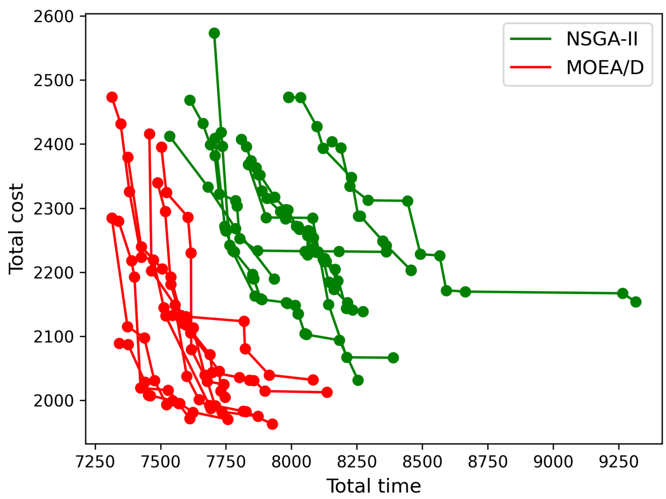


Figure 5.14: Plot of NSGA-II and MOEA/D solutions for instance 1 of 5_25_5 (N_TNOP_M).

Figure 5.14 illustrates the solutions obtained by NSGA-II and MOEA/D across ten runs for instance 1 of configuration 5_25_5. The visual comparison underscores MOEA/Ds dominance for this instance, with its solutions (red points) positioned closer to the ideal point than NSGA-IIs (green points). This superiority is quantified by MOEA/Ds significantly higher HV (1056) compared to NSGA-IIs (575). Moreover, the C-metric values further highlight **nsga-ii**'s dominance, with a value of 1 for MOEA/D (indicating none of its solutions are dominated by NSGA-II) and 0 for NSGA-II (indicating all its solutions are dominated by at least one MOEA/D solution).

The figure also reveals MOEA/Ds superior stability. Its solutions are tightly clustered, suggesting better convergence. In contrast, NSGA-II's solutions are more dispersed, indicative of inconsistency in solution quality.

This divergence in performance can be traced to the distinct exploration-exploitation strategies employed by the two algorithms. NSGA-II's larger population size (40), combined with a higher crossover probability (0.5) and mutation ratio (0.5), favours broad exploration of the search space. Conversely, MOEA/Ds smaller population size (20) and lower crossover (0.1) and mutation probabilities (0.2) focus on exploiting promising regions of the solution space.

The computational requirements explain their differing performance. NSGA-II's computationally intensive crowding distance calculation and large population size limit the number of generations it can complete. MOEA/D, with its simpler mechanisms and smaller population, allows for more generations, enabling deeper exploitation of the search space.

This aligns with the observed trend: NSGA-II tends to perform better on smaller problem instances or instances with non-conflicting objectives. In these cases, the search space is relatively constrained, and MOEA/Ds minor modifications to genes during mutation may have little to no impact due to the encoding mechanism. This can result in redundant solutions within the child population, which are subsequently filtered out. However, as problem size grows, NSGA-IIs performance declines due to its restricted number of iterations. Thus, MOEA/Ds advantage in larger instances may be attributed not only to its effective strategies but also to NSGA-IIs inherent limitations.

5.5 CONCLUSION

In this chapter, we address MPPP problem in the context of a RMS. The problem encompasses two subproblems:

- Process Planning: Assigning operations to machines, configurations, and tools while considering tool and TAD requirements, as well as sequencing the operations of each product.
- Product Sequencing: Determining the sequence of process plans for different products to minimize changeover time and cost between products.

The MPPP is formulated in a multi-objective framework with two objectives: minimizing the total production time and the total production cost.
To address this problem, we propose four solution approaches:

- Exact Methods: Two variations of NBI Scalarization technique are developed, each incorporating an iterative update function for the β parameter values. The first approach directly implements the NBI model in a solver, while the second employs an "enumerate-and-solve" strategy to enhance solution diversity and quality.
- Approximate Methods: Two evolutionary metaheuristics are utilized: NSGA-II and MOEA/D, both tailored to effectively handle the complexity of the problem.

The experimental findings reveal the strengths and limitations of the proposed approaches:

- Small Instances: All four methods effectively solve small-sized instances. Metaheuristics, however, outperform exact methods in limited computational time, producing high-quality solutions quickly.
- Medium to Large Instances: MOEA/D consistently outperforms NSGA-II across most evaluation metrics, demonstrating its efficiency in handling larger problem sizes.
- Computational Efficiency: Exact methods, NBI approaches, exhibit superior performance when provided with sufficient computational time, achieving better convergence to the true Pareto front.

The results further highlight that total cost is more sensitive to operation assignment decisions than to sequencing decisions. This finding suggests the potential for a hierarchical decision-making heuristic, where operation assignments are prioritized before sequencing. Such an approach could extend the heuristic framework proposed in Mechaacha et al., 2024, offering a practical means to achieve

high-quality solutions for complex MPPP problems.
This study underscores the applicability of both exact and approximate methods in process planning optimization and sets the stage for future exploration into hybrid or hierarchical frameworks to address the challenges posed by larger and more intricate RMS related problems.

6 LOT-SIZING WITH RMS

This chapter[1] presents the lot-sizing production planning problem with the consideration of reconfigurable machines. Firstly, the considered problem is described, along with the mathematical model for the uncapacitated case and its extension for the capacitated case. Next, a DP is proposed to solve the uncapacitated version. Finally, the results of the numerical experiments are presented and conclusions are drawn.

6.1 INTRODUCTION

The problem treated in this chapter considers lot-sizing problem in a shop floor composed of reconfigurable machines. As mentioned in chapter 1 these machines are different to traditional ones in that they can operate in alternative configurations, each with varying capacities or functionalities.

The objective is to determine the quantities to manufacture in a reconfigurable machine on a set of periods. In each period, a demand has to be satisfied, and a minimum batch size is imposed, thus for the manufacturer if production occurs in a period, it has to surpass a certain limit. The objective is to minimize the total cost comprised of production cost, inventory holding cost, startup costs, and reconfiguration cost of the machine, which represents the cost of changing the state of the machine—or the facility— from a current configuration to the next one.

For example, consider an assembly workstation with three configurations (Figure 6.1): C1 (fully manual), C2 (semi-manual), and C3 (fully automatic). Each configuration offers different throughputs and functionalities—C1 is good with high precision products but has low throughput and can assemble certain products, while C3 has a higher throughput but lower precision and can assemble some products that may or may not be manufacturable by the other two configurations.

The production costs also differ across configurations. For instance, C1's costs are largely driven by operator salaries, while C3's costs are influenced by factors like robot availability and maintenance. Setup costs vary as well, fluctuating over time based on factors such as labour costs, quality requirements, and energy prices. Consequently, different periods may favour different configurations—C1 for low labour costs and high-quality requirements, C2 or C3 for low energy costs and repetitive tasks.

In addition to the costs usually found in lot-sizing problems, we have *reconfiguration cost* or "*configuration changeover cost*", which depends on the sequence

1 Parts of this chapter will be submitted soon for possible publication in an international journal.

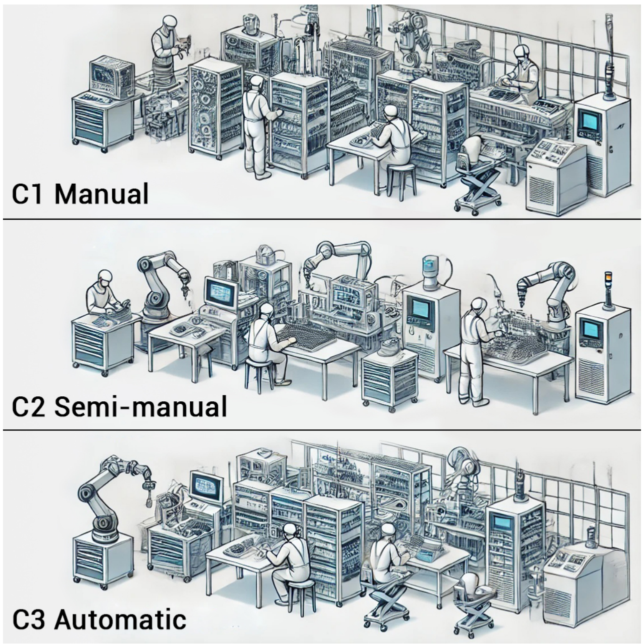


Figure 6.1: An imaginary illustration of a reconfigurable assembly workstation with three configurations C1 Manual, C2 Semi-manual, and C3 Automatic (Generated with AI).

of configurations being switched. A reader can detect directly the resemblance between this cost and sequence dependant cost which is employed in multi-item case. However, reconfiguration cost is configuration dependant and can be found in Single-item case.
The classical lot-sizing problem can be seen as a special case of the problem with reconfigurable machines, where the machine has only one fixed configuration. From this, we can derive a new class of lot-sizing problems with reconfigurable machines, which can be single or multi-item, capacitated or Uncapacitated, and single or multi-level.

Figure 6.2 illustrates a solution for the Uncapacitated SILSP with a single-level single-machine that ahas two possible configurations, across the two configurations the machine proposes two different processing speeds for processing the item. A configuration changeover time is incurred whenever the configuration changes between successive batches. Instead of setup carryover[2], we have—what we prefer to name—*configuration carryover*, which occurs from one period to the next.

Thus, when a machine uses the same configuration at the end of a period and the beginning of the next period, no configuration changeover cost or time is required. In the contrary, if two different configurations are used a reconfiguration cost is incurred.

2 Setup carryover refers to the condition where the setup state of a machine is maintained across consecutive periods, eliminating the need for a new setup at the start of the next period.

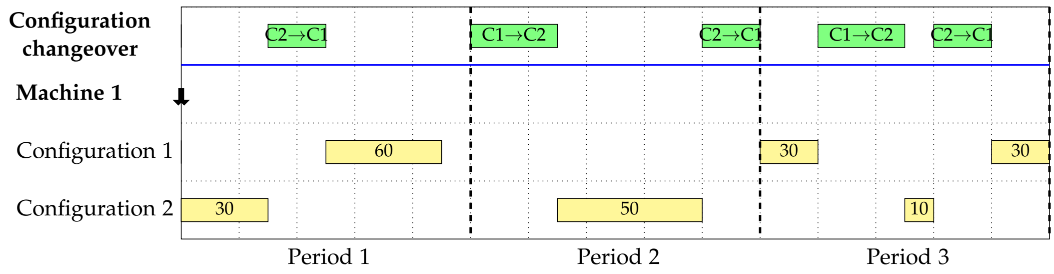


Figure 6.2: The Gantt chart corresponding to an Uncapacitated Single-item Single-level lot-sizing solution.

6.1.1 *Minimum batch size*

In the Uncapacitated SILSP presented in Karmarkar et al., 1987, they impose that in the absence of production, a setup (or reservation) can occur and thus the start-up can be avoided in the immediate subsequent period with production.
We believe this constraint should be relaxed as for multiple industries this case would not be possible, and such a solution wouldn't be applicable in real-life scenarios.
However, when running the model with the relaxed constraint, the solution may (in certain instances) contain periods of production where the production quantity is minimal but not zero. In such cases, the solver prefers to avoid a startup by setting up the machine for the production of a single unit, which can represent a very small value to the usual demand (for example 1 compared to 100). It is evident that this solution is not applicable in real-life scenarios either. Therefore, it is necessary to introduce a minimum batch size.
The minimum batch size can be defined as the minimum demand for all the periods. However, if the minimum demand is small compared to the usual demand (highly volatile demand), the model will exhibit the same behaviour previously described.

6.2 PROBLEM DESCRIPTION

We consider a SILSP with start-up costs and minimum batch sizes, referred to as MOQ. The production system is reconfigurable, which can be considered as a single machine, capable of operating in different configurations, each offering distinct processing times and costs for manufacturing the product. The MOQ is assumed to be equal to the demand in the given period ($MOQ_t = d_t$).
The company faces varying product demands across periods and incurs start-up, inventory holding, production, and reconfiguration costs. The manufacturer must determine the production quantity in each period to meet the demand while minimizing the total cost.

6.2.1 *Mathematical Models*

6.2.1.1 *Single-Item Uncapacitated Lot-sizing problem (P_0)*

To formulate the problem, the following notations are used:

Indexes	
t	Index of periods
c, e	Index of configurations

Parameters	
T	Number of Periods
C	Number of available configurations on the machine
d_t	demand of period t
h_t	Inventory cost for holding one unit of inventory in period t
rc_{ce}	Configuration changeover cost from configuration c to configuration e
pc_{ct}	Production cost on configuration c in period t
K_t	Startup cost in period t
M_{ct}	Upper bound on the quantity produced on configuration c in period t

Decision variables	
x_{ct}	Quantity produced on configuration c in period t
y_t	equals 1 if the machine is used in period t
z_t	equals 1 if a startup occurs on the machine in period t
w_{cet}	equals 1 if there's a configuration changeover from configuration c to configuration e in period t
I_t	Inventory level at the end of period t
V_{ct}	Auxiliary variables that assign configuration c in period t
U_{ct}	equals 1 if the machine is set up for configuration c at the beginning of period t

Objective function

$$\text{Minimize: } f_{cost} = PC + SC + CCC + IHC \tag{6.1}$$

$$\text{Processing cost:} \quad PC = \sum_c \sum_t pc_{ct} \times x_{ct} \tag{6.2}$$

$$\text{Startup cost:} \quad SC = \sum_t K_t \times z_t \tag{6.3}$$

$$\text{Configuration changeover cost:} \quad CCC = \sum_c \sum_e \sum_t rc_{ce} \times w_{cet} \tag{6.4}$$

$$\text{Inventory Holding cost:} \qquad IHC = \sum_t h_t \times I_t \tag{6.5}$$

Constraints

$$I_t = I_{t-1} + \sum_c x_{ct} - d_t \qquad \forall t = 1, ..., T, \quad \text{with: } I_0 = 0 \tag{6.6}$$

$$x_{ct} \leq M_{ct} \times (U_{ct} + \sum_e w_{ect}) \qquad \forall t = 1, ..., T, \quad \forall c = 1, .., C \tag{6.7}$$

$$x_{ct} \leq M_{ct} \times y_t \qquad \forall t = 1, ..., T, \quad \forall c = 1, .., C \tag{6.8}$$

$$z_t \geq y_t - y_{t-1} \qquad \forall t = 1, ..., T, \quad \text{with: } y_0 = 0 \tag{6.9}$$

$$\sum_c x_{ct} \geq y_t \qquad \forall t = 1, ..., T \tag{6.10}$$

$$\sum_c U_{ct} = 1 \qquad \forall t = 1, .., T \tag{6.11}$$

$$U_{ct} + \sum_e w_{ect} = U_{ct+1} + \sum_e w_{cet} \qquad \forall t = 1, .., T \quad \forall c = 1, .., C \tag{6.12}$$

$$V_{ct} + C \times w_{cet} - (C-1) - C \times U_{ct} \leq V_{et} \qquad \forall t = 1, .., T, \quad \forall c, e = 1, .., C, \&e \neq c \tag{6.13}$$

$$w_{cct} = 0 \qquad \forall t = 1, .., T, \quad \forall c = 1, .., C \tag{6.14}$$

$$\sum_c x_{ct} \geq y_t \times d_t \qquad \forall t = 1, .., T \tag{6.15}$$

$$y_t, z_t, w_{cet}, U_{cet} \in \{0, 1\} \qquad \forall t = 1, .., T \tag{6.16}$$

$$x_c t, I_t, V_{ct} \in \{0, 1\} \qquad \forall t = 1, .., T \tag{6.17}$$

In addition, let:

$$M_{ct} = \sum_{u=t}^{T} d_u \qquad \forall c = 1, .., C \tag{6.18}$$

The objective function (6.1) minimizes the sum of the four costs. Production cost is presented in equation (6.2) and is computed based on the decision variable x_{ct}. Startup costs are computed in (6.3) and are dependent on variable z_t. Reconfiguration costs in equation (6.4) are for computing the configuration changeovers, while (6.5) compute the inventory holding costs.

Equations (6.6) are the inventory balance equations, they enforce that the quantity produced in a period will be added to the inventory carried from the previous period and are used to satisfy the demand of that period while what's left is kept as inventory in the end of the period.

Constraints (6.7) indicates that when a machine is used in a configuration in a period, either that configuration was the start configuration or the machine has been reconfigured to that configuration.

Constraints (6.8) and (6.10) ensures that every period with a positive production is recorded through the y_t variable.

Constraints (6.9) implies that whenever the machine is used in a period without being used in the period that was before it a startup cost is added to the objective

function.
Constraints (6.11) state that only one configuration can be the first configuration to start a period.
Constraints (6.12) tracks the occurrence of configuration changeovers on the machine. Constraints (6.13) are used to eliminate subtours between configurations.
Constraints (6.14) states that there are no configuration changeovers from a given configuration to that exact same configuration.
Constraints (6.15) are MOQ constraints, which state that if production is positive in a period it has to be at least equal to the demand in that period. Constraints (6.16) and (6.17) are integrality constraints. Equations (6.18) are used to calculate the M parameters.
This problem from (6.1) to (6.18) will be refereed to in the remaining of this chapter as P_0.

6.2.1.2 *Single-Item Capacitated Lot-sizing problem*

We add the following parameters in the capacitated version:

Additional parameters	
r_{ce}	Configuration changeover time from configuration c to configuration e
p_{ct}	Production time on configuration c in period t
L_t	Length of period t

We keep the same indices, decision variables, and constraints, and we add the capacity constraints:

$$\sum_{c} p_{ct} \times x_{ct} + \sum_{c} \sum_{e} r_{ce} \times w_{cet} \leq L_t \qquad \forall t = 1, .., T \tag{6.19}$$

In addition, let:

$$M_{ct} = min(\sum_{u=t}^{T} d_u, \frac{L_t}{p_{ct}}) \qquad \forall c = 1, .., C \tag{6.20}$$

Constraints (6.19) represent capacity constraints on length of a period. While equations (6.20) replaces (6.18) from the P_0 model.

6.2.1.3 *Special case when $d_t = 0$*

Constraints (6.15) imply that if in a given period, $d_t = 0$, then no startup cost will be incurred in the subsequent period, even if $x_t = 0$, y_t would still equal to 1. We believe that this hypothesis remains consistent with reality. If the demand in a period is zero, it follows that the startup cost in the subsequent period is inevitable. In such a case, it would be reasonable to consider the startup cost as a fixed cost, which should not be included in the objective function. Furthermore, in certain industries, such as the production of frozen dairy products during winter months, periods of low demand are often used for essential cleaning and maintenance of machinery. In such cases, the shutdowns can be beneficial.

6.3 DP FOR SOLVING THE PROBLEM

This section introduces the DP approach developed to solve the problem P_0. It begins by establishing that the Wagner-Whitin (W-W) property, a key characteristic in lot-sizing problems, holds for the specific formulation of our problem. Following this foundational result, additional theorems and properties relevant to the problem are derived and discussed. These theoretical insights set the stage for the subsequent presentation of the dynamic programming model.

For developing a DP on the basis of DP of Wagner and Whitin, 1958 for P_0, one of the major necessities is to verify whether the problem does preserves the zero-inventory W-W property (Property 1 in Chapter 3 section 3.1.1).

6.3.1 *Zero inventory W-W Property for the Uncapacitated case*

The W-W property in the Uncapacitated case: $\sum_c x_{ct} \times I_t = 0$ needs to be verified, we propose the following theorem:

Theorem 1. *In P_0 there exists an optimal solution that satisfies the zero-inventory W-W property, i.e. $\sum_c x_{ct} \times I_t = 0$*

Proof. Let us consider the case where an optimal lot-sizing suggests both the carrying of inventory up to period j and the production in the same period. Consequently, there are two feasible scenarios that satisfy constraints (6.15). The first scenario is that the inventory carried is for satisfying demand in either the current period (j) or a subsequent period, which is equivalent to the original W-W case (Wagner and Whitin, 1958). The second scenario is that the demand for a period u ($j < u$) is divided between periods j and a preceding period i ($i < j$). In this case, at least one of the following programs wouldn't be more costly:

- To produce both parts of d_u in period i and thus $\sum_c x_{ci} = \sum_{k=i}^{u} d_k$
- To produce both parts in period j thus having $\sum_c x_{cj} = \sum_{k=j}^{u} d_k$
- To produce both parts in period u which would result in $\sum_c x_{cu} = d_u$

□

6.3.2 *Lot-splitting*

Lot-splitting in lot-sizing problems refers to producing the same demand for an item in two different periods. We extend this definition for our case, where it refers to producing the same item in two different configurations within the same period.

6.3.2.1 *Lot-splitting for the Single-item Capacitated case*

Let's take the following example with **1 period** and **2 configurations**. The first configuration C1 is slow but cheap, C2 is fast but expensive. and the data is as follows: demand $d_1 = 10$, Capacity $L_t = 14$, production times $p_{ct} = [3 \quad 1]$, production costs $pc_{ct} = [1 \quad 10]$. All reconfiguration times are equal to 2, and reconfiguration costs are equal to 3.

Figure 6.3 represent the Gantt chart of the optimal solution with the objective function calculation at the top (94 \$). As it can be seen from the Figure the op-

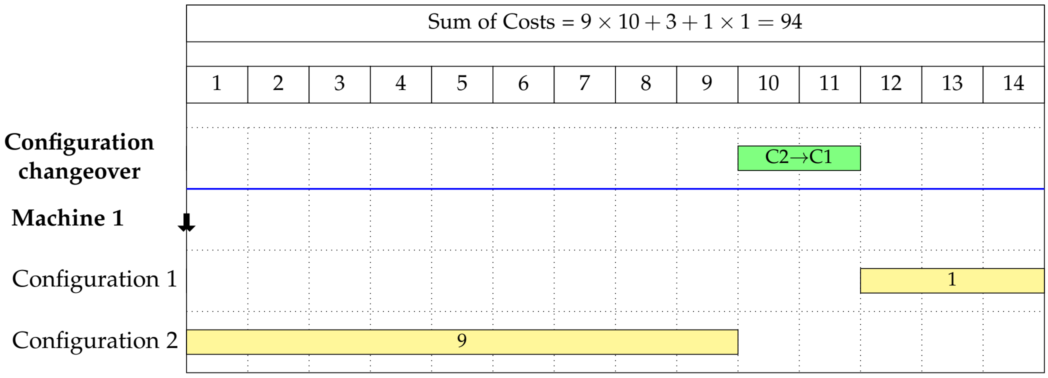


Figure 6.3: The Gantt chart corresponding to the optimal solution for the Capacitated Single-item case with Lot-splitting.

timal solution for this particular instance contains lot-splitting. Thus in the case of capacitated case the optimal solution can contain lot-splitting.

6.3.2.2 *Lot-splitting for the Single-item Uncapacitated (P_0 case)*

Theorem 2. *There exists an optimal solution in P_0, where no lot-splitting occurs, i.e.* $\sum_c y_{ct} = 1 \quad \forall t = 1, .., T$ *with:*

$$y_{ct} = \begin{cases} 1 & \textit{if } x_{ct} > 0 \\ 0 & \textit{if } x_{ct} = 0. \end{cases} \tag{6.21}$$

Proof. Suppose an optimal program suggests both to produce in configurations c and e in the same period t. Then, in the case where $pc_{ct} \leq pc_{et}$ it wouldn't be more costly to add x_{et} into x_{ct}, since this would eliminate the reconfiguration cost rc_{cet} (or rc_{ect}), and reduce the production cost if $pc_{ct} < pc_{et}$. In the opposite case ($pc_{ct} \geq pc_{et}$), it wouldn't be more costly to add x_{ct} into x_{et} either. □

6.3.3 *Reconfiguration*

For the configuration changeover on the machine, we propose the following theorem:

Theorem 3. *In P_0 with constant reconfiguration costs (rc_{ce} doesn't vary from a period to the other), there exists an optimal solution where configuration changeovers occur only in periods with production (where $\sum_c x_{ct} > 0$).*

Proof. Suppose an optimal plan suggests to not to produce and change the configuration in the same period t. If there's no period q with production ($\sum_c x_{cq} > 0$) after t ($q > t$) that would mean that the configuration changeover is unnecessary and thus can be eliminated. If not, that would mean that the directly successive period with production after t (name it q) will use the changed configuration c from period t, thus we have—as a result of theorem 2— $x_{cq} > 0$. As the reconfiguration costs satisfy the triangle inequality, are not time varying, and as the problem is Uncapacitated, it wouldn't be more costly to move the reconfiguration from period t to the beginning of period q. □

6.3.4 *DP*

Theorem 4. *There is an $O(T^2 \cdot C)$ algorithm that solves P_0 (Uncapacitated SILSP with minimum batch size and constant reconfiguration costs).*

A dynamic programming approach, similar to the W-W algorithm and employing Karmarkar et al., 1987 theorems, can be employed to efficiently solve the Uncapacitated problem, with the additional consideration of the optimal configuration of the machine in each period.
Starting from the first period the problem is solved just as the W-W case but with consideration of all the cases of the configurations and verifying as in Karmarkar et al., 1987 DP that the startup cost is added to the next period's subproblem. That's why the computational complexity of DP is $O(T^2 \cdot C)$.

6.4 NUMERICAL EXPERIMENTS

This section outlines first the data generation details, next the results are presented for the Uncapacitated problem. The DP algorithm was coded in Java programming language and the ILP model is solved using CPLEX 22.1 version, the tests are conducted on a PC with a 1.20GHz GHz Intel i5 processor and 8GB RAM.

We defined 14 instance configurations and generated 10 random instances per configuration (140 instances in total) with a uniform distribution: $d \in \{0, 300\}$, $h \in \{0, 5\}$, $pc \in \{0.5, 20\}$, $K \in \{30, 500\}$, and for rc parameter we used the generation procedure proposed in Appendix A with the following values $rc \in 20; 200, 200$. And we set the CPU limit to 1800 seconds for the ILP.

The gaps from DP solution to best solution (UB) of the solver were calculated using the formula:

$$\text{Gap UB to DP} = \frac{\text{Solver UB} - \text{DP solution}}{\text{DP solution}} \times 100$$

The integrality gap[3] (Int Gap) obtained by the solver was also recorded. The integrality gap is computed as

$$\text{Int Gap} = \frac{\text{UB} - \text{LB}}{\text{UB}} \times 100$$

Table 6.1 presents the results obtained across the 14 instance configurations (instance sizes). An instance configuration is characterised by two parameters: the number of periods (T) and the number of configurations of the reconfigurable machine (C). As previously mentioned, the DP CPU depends on these two parameters. For example, instance T10C3 represents a facility with 3 possible configurations and the planning horizon is composed of 10 periods.

Table 6.1: Results obtained for comparing the DP with ILP mathematical model implemented in CPLEX solver.

Instance	Gaps		CPU Time	
	Int Gap (%)	Gap UB to DP (%)	DP (sec)	CPLEX (sec)
T5C3	0.000	0.000	0.006	0.122
T5C5	0.000	0.000	0.002	0.095
T10C3	0.000	0.000	0.002	0.087
T10C5	0.001	0.000	0.002	0.245
T10C10	0.002	0.000	0.004	1.807
T10C20	0.005	0.000	0.007	16.830
T50C10	0.150	0.000	0.048	253.961
T50C30	5.296	0.129	0.047	1806.715
T50C50	8.510	1.693	0.083	1803.870
T50C100	91.165	817.008	0.264	1803.678
T100C30	6.590	0.600	0.251	1811.468
T100C50	30.545	99.970	0.609	1809.857
T100C100	93.606	841.257	1.488	1810.103
T200C200	NO Sol	NO Sol	47.300	NO Sol
Average	18.144	135.435	0.216	855.295

As can be observed from Table 6.1 results, the DP significantly outperforms the solver. The DP provides the optimal solution in a remarkable average time of less than a quarter of a second (0.21 second) across all configurations. In the case of the seven smallest configurations, the solver is able to identify the optimal solution that is consistent with those produced by the DP. However, for only the three smallest configurations that the solver closes the integrality gap to 0%, while the solver does reach the optimal solution, it is unable to guarantee that this is the optimal solution. For example for T50C10 in 3 out of the 10 generated instances

3 The integrality gap is the ratio between the best value and best lower bound.

the integrality gap was not close. Consequently, it can be seen that the average integrality gap for T50C10 is 0.150.
Once the instances sizes exceeds T50C30, the solver is unable to find the optimal solution within the 1800 seconds limit, not even without guaranteeing its optimality. As the configuration sizes increase, the solver's performance deteriorates significantly, resulting in UB that are 8 times distant from the optimal solution within 1800 seconds for T100C100.
The mathematical model appears to encounter difficulties, particularly in instances with a high number of configurations. Nevertheless, the increase in the number of periods still affects the ILP model, but the increase in C has a more pronounced impact as can be seen between T50C100 and T100C50 results.

6.5 CONCLUSION

In conclusion this chapter have presented the problem of single-item lot-sizing problem with startup costs and time variant MOQ with a RMS. The problem has been mathematically modelled in its two variants capacitated and Uncapacitated by ILP models. Our main focus was on the Uncapacitated problem (P_0) where we developed a DP for solving the problem. The proposed ILP model for P_O was implemented in a solver (Cplex) and solved 140 random generated instances and compared to DP performance.
The DP largely outperformed the mathematical model especially for large size instances, thereby showcasing its applicability in different lot-sizing parameter values. This provides a solid foundation for further investigation of more complex scenarios, by integrating the DP within a Lagrangian relaxation approach or a branch-and-bound algorithm. Although less effective than DP, the two ILPs are indispensable for validating solution approaches and solving multiple configurations that are inherently challenging. In particular, those the model for the capacitated variant, which was not tested in this chapter but will be included as a part in the future work which will be soon submitted for publication.

7 CONCLUSIONS AND FUTURE WORK

The contributions of this thesis can be summarized in three major points:

- Introduction of product sequencing into the process planning problem with RMS.
- Study of the lot-sizing problem with reconfigurable machines.
- The deployment of diverse methods for resolving the proposed problems, encompassing both mono-objective (DP, lower bounds, mathematical-based heuristics, mathematical models) and multi-objective (NSGA-II, MOEA/D, NBI with β updated values) optimisation.

In this concluding chapter, we present a summary of our contributions and provide a list of research perspectives, both short-term and long-term, which we intend to develop following the completion of this thesis. These perspectives may also be considered by other researchers.

This thesis addresses three major research topics, each of which represents a production planning problem with RMS:

- Motivated by the extensive body of literature addressing process planning problems with RMS, We adopted an alternative standpoint to the classical process planning works in addressing the problem, focusing on methodological aspects while maintaining an understanding of the technical elements. This approach has yielded the development of an effective mathematical model, which reduces the computational time, and a mathematical-based heuristic with two variants, which can be used as a framework for solving other variants of the problem. Furthermore, our work has resulted in the development of two lower bounds for the total time objective function.
- Once the fundamentals had been established, we proceeded to integrate product sequencing, which is a scheduling decision, into the process planning problem, which we termed "Multi-product process planning." This problem was studied in a multi-objective context, with the formulation of a mathematical model that was employed in a normal-boundary intersection approach and an enumerative approach, in addition to two evolutionary metaheuristics, namely MOEA/D and NSGA-II. In order to evaluate the aforementioned methods, a number of multi-objective performance indicators were employed, including HV and C-metric.

- Ultimately, as our comprehension of the reconfigurability paradigm and RMS was enhanced, We focused our attention on a relatively unexplored area within the realm of RMS: lot sizing with RMS. This relatively novel problem was studied in a single-item context, with a particular focus on defining the problem constraints and ensuring coherence with existing literature on both RMS and lot-sizing. We employed a highly regarded methodology in lot sizing, namely dynamic programming. The work yielded the development of mathematical models for lot-sizing with a reconfigurable machine and a single item for both capacitated and Uncapacitated variants, as well as the development of a DP for the Uncapacitated case.

The proposed approaches are applicable within an e-CAPP framework. The LBs can verify the existence of a feasible[1] process plan and estimate its total production time, and has the potential to be extended for estimating energy consumption and production cost. Timely access to such production process insights can significantly enhance product design effectiveness and facilitate web-based collaboration, ultimately improving manufacturing responsiveness. The heuristics are also well-suited for collaborative environments where a cloud-based knowledge repository would enable knowledge sharing between different entities of the company and between companies. This fosters mutual benefit by accessing each other's accumulated expertise. However, a standardized format for representing process plans, especially for RMTs with diverse module combinations, is crucial for successful implementation. The knowledge repository can also be used by AI techniques in a knowledge-based process planning. This represent an interesting perspective where the heuristic proposed in chapter 4 is used for solving the MPPP problem proposed in chapter 5.

Process planning problem within a hybrid system composed of DML, FMS, and RMS combination can be also an interesting problem related perspective. It is possible to introduce further constraints into the process planning and the production planning problems studied, such as machine temperature. In such cases, the quality of the part produced is dependent on the temperature of the machine (for further details on this, please see Khan et al., 2021, which provides an example of quality in process planning).

One limitation with our proposed approaches (especially for chapter 4 and 5) is that our models cannot handle alternative operation selection, where a part can be processed by multiple operation combinations. In such a case, a planner has to choose which operations to execute before assigning them to machines and performing the sequencing. Such a constraint can improve the realism of the process planning problem considered and further explore the potential of RMTs.

Another potential future work is to extend the methods proposed for SUPP to handle multi-objective problems, other objectives such as cost, quality and flexibility can be included and optimized. Moreover, it would be interesting to compare the performance of the proposed methods against other popular evolutionary algorithms such as GA (for the mono-objective case), and NSGA-III (for the multi-objective case). Also, to demonstrate the applicability and robustness of the pro-

1 Feasibility here is only in terms TADs and tools constraints, since the model doesn't consider precedence constraints.

posed methods, an empirical evaluation into real-world manufacturing settings or more complex scenarios should be performed, however , to the best of our knowledge no databases for such instances on RMS process planning or RMS lot-sizing exist in the literature.

As already demonstrated in chapter 5, the disruptive nature of the encoding procedure and genetic operators provided MOEA/D with an advantage over NSGA-II. Therefore, to enhance the performance of NSGA-II and other evolutionary metaheuristics, alternative encoding schemes and genetic operators tailored to the specific characteristics of the MPPP problem could be investigated.

The proposed approaches for integration between process and production planning, can be integrated into an RMS model (See André and Cardin 2023), where a dedicated software module is designed for process and production planning. This package considers various factors related to machine configurations and operational requirements. The availability of multiple approaches allows for the selection of the most suitable model, taking into account factors such as available CPU time and the weight of each objective, providing decision makers with multiple trade-off solutions for optimal decision-making in RMS operations.

As RMS has been recognized as a facilitator of sustainable manufacturing (Skärin et al. 2022), incorporating environmental objectives, such as minimizing greenhouse gas emissions and energy consumption, and social sustainability ones into the the studied optimization problems can be a promising direction, which is currently receiving an increasing attention.

To explore the potential of other metaheuristic paradigms, swarm intelligence-based approaches such as Multi-Objective Ant Colony Optimization could be investigated. Additionally, a comparative study between the proposed operation position-based decision variables model (1MDV) and a sequence-based mathematical model would provide valuable insights into the efficiency of different problem representations, for SUPP and MPPP problems.

The results also indicate that the total cost is more sensible to operations assignment decisions than sequencing decisions. Suggesting that a hierarchical decision-making heuristic can provide good quality solutions. Such an approach could be developed as an extension of the heuristic framework we proposed in Mechaacha et al., 2024 and chapter 4.

With the emergence of human worker considerations, workforce planning with RMS have gained increased attention (Dolgui et al. 2019). Thus, another future extension can incorporate these aspects, introducing worker assignment to operations. The operation-worker assignment would depend on different factors such as operation difficulty and worker proficiency. This can take use of the results of the data-driven analysis presented in Lee et al., 2023. Which represents also an interesting extension to all our three projects (SUPP, MPPP, and SILSP).

Furthermore, a sensitivity analysis studying the impact of varying the processing and changeover times/costs on solution structure and objective function values, similar to S. Zhang and Wong, 2016, but considering MPPP, would be an interesting work and provide valuable insights into the problem's behaviour.

Finally, an interesting extension would be to consider Distributed Multi-Product Process Planning (D-MPPP), where products are not only sequenced but also allocated to different production sites (Mahmoodjanloo et al. 2022). This would introduce an additional decision layer (site assignment) that could enhance a company's ability to adapt to volatile market demands by leveraging the reconfigurability of different

It is important to recognise that chapter 6 work represents only the initial stage of an ongoing process, as the introduction of RMS to lot-sizing has opened up a new area of lot-sizing problems. A natural extension to this work is the consideration of multi-item cases in the short term. While DP is an effective approach, it would be beneficial to explore alternative methods for NP-hard problems, such as Lagrangian relaxation and branch-and-bound.

As the problem under consideration is a special case of the MOQ, namely where the MOQ is equal to the demand, a natural extension would be to consider a more general case with a fixed MOQ as in Hellion et al., 2012. Subsequently, the more complex problem of general time-varying MOQs can be addressed. While the development of DPs may present a challenge in these cases, our DP can serve as a foundation for that process or, combined with a correction heuristics in a Lagrangian relaxation method, facilitate the generation of good-quality solutions.

Additionally, investigating multi-level lot-sizing with reconfigurable machines is a natural extension for the single level case. The integration of process planning and lot-sizing represents a long-term yet promising research avenue for the future.

A

APPENDIX 1: CHANGEOVER MATRICES GENERATION PROCEDURE (TRIANGLE INEQUALITY)

To obtain an asymmetric matrix that satisfies the conditions of having zero diagonal entries and approximates the ranges, means, and standard deviations of the matrices in Haddou Benderbal et al., 2018 (which do not satisfy the triangle inequality), two matrices are generated and their sum is calculated.

Matrix A is generated through a uniform map-based approach that is implemented based on the method described in Pickardt and Branke, 2012. This method involves the distribution of a set of points uniformly in a predefined space, followed by the computation of the Euclidean distance between every pair of points x_1 and x_2, where the distance corresponds to the changeover time from x_1 to x_2.

Matrix A satisfies the triangle inequality as a consequence of the Pythagorean theorem. However, the matrix generated is symmetric, which fails to represent situations where the setup time depends on the direction of the changeover, as in the problems treated in chapters: 4, 5, and 6. Additionally, some distances might have a value of zero if the coordinates of x_1 and x_2 are the same.

Hence, a second matrix B is generated using the uniform distribution U(min, max) where $max = (min \times 2) - 1$. Matrix B satisfies the triangle inequality as the following equation holds: $min \times n \geq max$ for all $n > 1$. Additionally, B is asymmetric.

It should be noted that matrix B alone cannot be used, as the range of its entries is fixed with a ratio of $\frac{max(b_{i,j})}{min(b_{i,j})} \leq 2$, while in Haddou Benderbal et al., 2018, this ratio can reach up to 10 in some cases. Therefore, the obtained matrix C is the sum of matrices A and B, i.e., $A + B = C$.

proof that C satisfies the triangle inequality:
Let A and B be two matrices that satisfy the triangle inequality. where:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \qquad B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix} \tag{A.1}$$

Since A and B satisfy the triangle inequality then:

$$a_{i,k} + a_{k,j} \geq a_{i,j} \qquad \forall i, j, k \in 1, \cdots, n \tag{A.2}$$

$$b_{i,k} + b_{k,j} \geq b_{i,j} \qquad \forall i, j, k \in 1, \cdots, n \tag{A.3}$$

Let matrix C be the sum of A and B. where:

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ \vdots & \vdots & \ldots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix} \tag{A.4}$$

For C to satisfy the triangle inequality, we must have:

$$c_{i,k} + c_{k,j} \quad \geq \quad c_{i,j} \qquad \forall i, j, k \quad \in \quad 1, \cdots, n \tag{A.5}$$

which means:

$$a_{i,k} + b_{i,k} + a_{k,j} + b_{k,j} \geq a_{i,j} + b_{i,j} \qquad \forall i, j, k \in 1, \cdots, n \tag{A.6}$$

condition (A.6) is already satisfied since A and B satisfy the triangle inequality.

BIBLIOGRAPHY

Abbasi, M., & Houshmand, M. (2011). Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *54*(1), 373–392.

Absi, N., Dauzère-Pérès, S., Kedad-Sidhoum, S., Penz, B., & Rapine, C. (2016). The single-item green lot-sizing problem with fixed carbon emissions. *European Journal of Operational Research*, *248*(3), 849–855.

Aguilar, A., Roman-Flores, A., & Huegel, J. C. (2013). Design, refinement, implementation and prototype testing of a reconfigurable lathe-mill. *Journal of manufacturing systems*, *32*(2), 364–371.

Aksen, D., Altnkemer, K., & Chand, S. (2003). The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, *147*(3), 558–566.

Ameer, M., & Dahane, M. (2023). Reconfigurability improvement in industry 4.0: A hybrid genetic algorithm-based heuristic approach for a co-generation of setup and process plans in a reconfigurable environment. *Journal of Intelligent Manufacturing*, *34*(3), 1445–1467.

Andersen, A.-L., Andersen, R., Napoleone, A., Brunø, T., Kjeldgaard, S., Nielsen, K., Sorensen, D., Raza, M., Bilberg, A., Rösiö, C., Boldt, S., & Skärin, F. (2023, May). *Paving the way for changeable and reconfigurable production: Fundamental principles, development method & examples* (1st ed.). REKON Press.

Andersen, A.-L., Brunoe, T. D., & Nielsen, K. (2015). Reconfigurable manufacturing on multiple levels: Literature review and research directions. *IFIP International conference on advances in production management systems*, 266–273.

Andersen, A.-L., Rösiö, C., Bruch, J., & Jackson, M. (2016). Reconfigurable manufacturing–an enabler for a production system portfolio approach. *Procedia CIRP*, *52*, 139–144.

Anderson, E. J., & Cheah, B. S. (1993). Capacitated lot-sizing with minimum batch sizes and setup times. *International Journal of Production Economics*, *30*, 137–152.

André, P., & Cardin, O. (2023). A core reference model for applicable reconfigurable manufacturing systems. *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 507–519.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., & Salomon, L. (2021). Performance indicators in multiobjective optimization. *European journal of operational research*, *292*(2), 397–422.

Azab, A., & ElMaraghy, H. (2007a). Mathematical modeling for reconfigurable process planning. *Cirp Annals*, *56*(1), 467–472.

Azab, A., & ElMaraghy, H. (2007b). Sequential process planning: A hybrid optimal macro-level approach. *Journal of Manufacturing Systems*, *26*(3-4), 147–160.

Battaa, O., Dolgui, A., & Guschinsky, N. (2017). Decision support for design of reconfigurable rotary machining systems for family part production. *International Journal of Production Research*, *55*(5), 1368–1385.

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, *60*(6), 503–515.

Bellman, R. (1966). Dynamic programming. *science*, *153*(3731), 34–37.

Benderbal, H. H., Dahane, M., & Benyoucef, L. (2015). A new robustness index formachines selection in reconfigurable manufacturing system. *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, 1019–1026.

Benderbal, H. H., Dahane, M., & Benyoucef, L. (2018). Exhaustive search based heuristic for solving machine layout problem in reconfigurable manufacturing system design. *IFAC-PapersOnLine*, *51*(11), 78–83.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2011). Simulation-based nsga-ii approach for multi-unit process plans generation in reconfigurable manufacturing system. *ETFA2011*, 1–8.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2013a). Amosa based approach for process plan generation in reconfigurable manufacturing system with machine load balancing. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 958–963.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2014a). A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. *International Journal of Production Research*, *52*(12), 3583–3594.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2013b). A simulation-based genetic algorithm approach for process plans selection in uncertain reconfigurable environment. *IFAC Proceedings Volumes*, *46*(9), 1961–1966.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2012). Process plan generation in reconfigurable manufacturing systems using amosa and topsis. *IFAC Proceedings Volumes*, *45*(6), 560–565.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2013c). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, *66*(3), 519–524.

Bensmaine, A., Dahane, M., & Benyoucef, L. (2014b). Design of reconfigurable manufacturing systems: Optimal machines selection using nondominated sorting genetic algorithm (nsga-ii). *41st international conference on computers & industrial engineering*, 110–115.

Bensmaine, A. (2013). *Algorithmes évolutionnaires et méthodes approchées multicritères pour la génération des processus de fabrication dans un environnement reconfigurable* [Doctoral dissertation, Université de Lorraine].

Bortolini, M., Galizia, F. G., & Mora, C. (2018). Reconfigurable manufacturing systems: Literature review and research trend. *Journal of manufacturing systems*, *49*, 93–106.

Brahimi, N. (2004). *Planification de la production: Modèles et algorithmes pour les problèmes de dimensionnement de lots* [Doctoral dissertation].

Brahimi, N., Absi, N., Dauzère-Pérès, S., & Nordli, A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, *263*(3), 838–863.

Brahimi, N., Dauzere-Peres, S., Najid, N. M., & Nordli, A. (2006). Single item lot sizing problems. *European Journal of Operational Research*, *168*(1), 1–16.

Brahimi, N., Dolgui, A., Gurevsky, E., & Yelles-Chaouche, A. R. (2019). A literature review of optimization problems for reconfigurable manufacturing systems. *IFAC-PapersOnLine*, *52*(13), 433–438.

Campos Sabioni, R., Daaboul, J., & Le Duigou, J. (2022). Concurrent optimisation of modular product and reconfigurable manufacturing system configuration: A customer-oriented offer for mass customisation. *International journal of production research*, *60*(7), 2275–2291.

Chaube, A., Benyoucef, L., & Tiwari, M. K. (2012). An adapted nsga-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. *Journal of Intelligent Manufacturing*, *23*(4), 1141–1155.

Chryssolouris, G. (2013). *Manufacturing systems: Theory and practice*. Springer Science & Business Media.

Chugh, T., Gaspar-Cunha, A., Deutz, A., Duro, J., Oara, D., & Rahat, A. (2023). Identifying correlations in understanding and solving many-objective optimisation problems. In *Many-criteria optimization and decision analysis: State-of-the-art, present challenges, and future perspectives* (pp. 241–267). Springer.

Dahane, M., & Benyoucef, L. (2016). An adapted nsga-ii algorithm for a reconfigurable manufacturing system (rms) design under machines reliability constraints. In *Metaheuristics for production systems* (pp. 109–130). Springer.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, *8*(3), 631–657.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Dolgui, A., Petroodi, S. E. H., Kovalev, S., Kovalyov, M. Y., & Thevenin, S. (2019). Workforce planning and assignment in mixed-model assembly lines as a factor of line reconfigurability: State of the art. *IFAC-PapersOnLine*, *52*(13), 2746–2751.

Dou, J., Dai, X., & Meng, Z. (2011). A ga-based approach for optimizing single-part flow-line configurations of rms. *Journal of Intelligent Manufacturing*, *22*(2), 301.

ElMaraghy, H. A. (1993). Evolution and future perspectives of capp. *CIRP annals*, *42*(2), 739–751.

Fan, J., Zhang, C., Shen, W., & Gao, L. (2023). A matheuristic for flexible job shop scheduling problem with lot-streaming and machine reconfigurations. *International Journal of Production Research*, *61*(19), 6565–6588.

Fan, J., Zhang, C., Yang, F., Shen, W., & Gao, L. (2024). A matheuristic with re-lot-sizing strategies for flexible job-shop rescheduling problem with lot-streaming and machine reconfigurations. *European Journal of Operational Research*.

Fang, H., Wang, Q., Tu, Y.-C., & Horstemeyer, M. F. (2008). An efficient non-dominated sorting method for evolutionary algorithms. *Evolutionary computation*, *16*(3), 355–384.

Furlan, M., Almada-Lobo, B., Santos, M., & Morabito, R. (2024). Matheuristic for the lot-sizing and scheduling problem in integrated pulp and paper production. *Computers & Industrial Engineering*, *192*, 110183.

Gadalla, M., & Xue, D. (2017). Recent advances in research on reconfigurable machine tools: A literature review. *International Journal of Production Research*, *55*(5), 1440–1454.

Gao, S., Daaboul, J., & Le Duigou, J. (2021). Process planning, scheduling, and layout optimization for multi-unit mass-customized products in sustainable reconfigurable manufacturing system. *Sustainability*, *13*(23), 13323.

Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics* (Vol. 2). Springer.

Giglio, D., Paolucci, M., & Roshani, A. (2017). Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems. *Journal of cleaner production*, *148*, 624–641.

Gilchrist, A. (2016). *Industry 4.0: The industrial internet of things*. Springer.

Gu, P., & Norrie, D. H. (1995). *Intelligent manufacturing planning*. Springer.

Gu, X., & Koren, Y. (2018). Manufacturing system architecture for cost-effective mass-individualization. *Manufacturing letters*, *16*, 44–48.

Gu, X., & Koren, Y. (2024). Smart factories for mass individualization. *Encyclopedia*, *4*(1), 415–429.

Gumasta, K., Kumar Gupta, S., Benyoucef, L., & Tiwari, M. (2011). Developing a reconfigurability index using multi-attribute utility theory. *International Journal of Production Research*, *49*(6), 1669–1683.

Haase, K. (2012). *Lotsizing and scheduling for production planning* (Vol. 408). Springer Science & Business Media.

Haddou Benderbal, H., Dahane, M., & Benyoucef, L. (2017). Flexibility-based multi-objective approach for machines selection in reconfigurable manufacturing system (rms) design under unavailability constraints. *International Journal of Production Research*, *55*(20), 6033–6051.

Haddou Benderbal, H., Dahane, M., & Benyoucef, L. (2018). Modularity assessment in reconfigurable manufacturing system (rms) design: An archived multi-objective simulated annealing-based approach. *The International Journal of Advanced Manufacturing Technology*, *94*(1), 729–749.

Haddou-Benderbal, H., Dahane, M., & Benyoucef, L. (2016). Hybrid heuristic to minimize machines unavailability impact on reconfigurable manufacturing system using reconfigurable process plan. *IFAC-PapersOnLine*, *49*(12), 1626–1631.

Haddou-Benderbal, H., Dahane, M., & Benyoucef, L. (2017). Layout evolution effort for product family in reconfigurable manufacturing system design. *IFAC-PapersOnLine*, *50*(1), 10166–10171.

He, L., Chiong, R., Li, W., Budhi, G. S., & Zhang, Y. (2022). A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles. *Knowledge-Based Systems*, *243*, 108315.

Heemskerk, C. J. M., Reijers, L., & Kals, H. (1990). A concept for computer-aided process planning of flexible assembly. *CIRP annals*, *39*(1), 25–28.

Hellion, B., Mangione, F., & Penz, B. (2012). A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs. *European Journal of Operational Research*, *222*(1), 10–16.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM journal on computing*, *2*(2), 88–105.

Hopp, W. J., & Spearman, M. L. (2011). *Factory physics*. Waveland Press.

Ishii, N., Fuchino, T., & Muraki, M. (1997). Life cycle oriented process synthesis at conceptual planning phase. *Computers & chemical engineering*, *21*, S953–S958.

Jiang, B., Kang, Y., & Zhang, C. (2024). Two-stage robust optimization for lot-sizing and scheduling problems considering machine reconfiguration. *Proceedings of the 2024 6th International Conference on Management Science and Industrial Engineering*, 301–308.

Kant, R., Pattanaik, L. N., & Pandey, V. (2020). Sequential optimisation of reconfigurable machine cell feeders and production sequence during lean assembly. *International Journal of Computer Integrated Manufacturing*, *33*(1), 62–78.

Karmarkar, U. S., Kekre, S., & Kekre, S. (1987). The dynamic lot-sizing problem with startup and reservation costs. *Operations Research*, *35*(3), 389–398.

Katz, R. (2007). Design principles of reconfigurable machines. *The International Journal of Advanced Manufacturing Technology*, *34*, 430–439.

Kazemisaboor, A., Aghaie, A., & Salmanzadeh, H. (2022). A simulation-based optimisation framework for process plan generation in reconfigurable manufacturing systems (rmss) in an uncertain environment. *International Journal of Production Research*, *60*(7), 2067–2085.

Khan, A. S., Homri, L., Dantan, J. Y., & Siadat, A. (2021). Modularity-based quality assessment of a disruptive reconfigurable manufacturing system-a hybrid meta-heuristic approach. *The International Journal of Advanced Manufacturing Technology*, *115*(5-6), 1421–1444.

Khan, A. S., Homri, L., Dantan, J. Y., & Siadat, A. (2022). An analysis of the theoretical and implementation aspects of process planning in a reconfigurable manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 1–32.

Khettabi, I., Benyoucef, L., & Amine Boutiche, M. (2022). Sustainable multi-objective process planning in reconfigurable manufacturing environment: Adapted new dynamic nsga-ii vs new nsga-iii. *International Journal of Production Research*, *60*(20), 6329–6349.

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

Khezri, A., Benderbal, H. H., & Benyoucef, L. (2021). Towards a sustainable reconfigurable manufacturing system (srms): Multi-objective based approaches for process plan generation problem. *International Journal of Production Research*, *59*(15), 4533–4558.

Khezri, A., Benderbal, H. H., Benyoucef, L., & Dolgui, A. (2020). Diagnosis on energy and sustainability of reconfigurable manufacturing system (rms) design: A bi-level decomposition approach. *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 141–145.

Klingenberg, C. O., Borges, M. A. V., & do Vale Antunes Jr, J. A. (2022). Industry 4.0: What makes it a revolution? a historical framework to understand the phenomenon. *Technology in Society*, *70*, 102009.

Koren, Y. (2010). *The global manufacturing revolution: Product-process-business integration and reconfigurable systems*. John Wiley & Sons.

Koren, Y., Gu, X., & Guo, W. (2018). Reconfigurable manufacturing systems: Principles, design, and future trends. *Frontiers of Mechanical Engineering*, *13*(2), 121–136.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & Van Brussel, H. (1999). Reconfigurable manufacturing systems. *CIRP annals*, *48*(2), 527–540.

Larsen, N. E. (1993). Methods for integration of process planning and production planning. *International Journal of Computer Integrated Manufacturing*, *6*(1-2), 152–162.

Lee, C., Yun, J., Kim, G.-Y., Lim, J., Do Noh, S., & Kim, Y. (2023). Data-driven analysis and assignment of manual assembly production lines. *IFIP International Conference on Advances in Production Management Systems*, 535–547.

Li, H., & Zhang, Q. (2008). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE transactions on evolutionary computation*, *13*(2), 284–302.

Li, L., Klabjan, D., & Pun, C. (2011). Single item lot-sizing problem with minimum order quantity.

Li, P., Xue, Q., Zhang, Z., Chen, J., & Zhou, D. (2023). Multi-objective energy-efficient hybrid flow shop scheduling using q-learning and gvns driven nsga-ii. *Computers & Operations Research*, *159*, 106360.

Li, W., & McMahon, C. A. (2007). A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, *20*(1), 80–95.

Love, S. F. (1973). Bounded production and inventory models with piecewise concave costs. *Management Science*, *20*(3), 313–318.

Luo, K., Shen, G., Li, L., & Sun, J. (2022). 0-1 mathematical programming models for flexible process planning. *European Journal of Operational Research*.

Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., & Bozorgi-Amiri, A. (2020). Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm. *Applied Soft Computing*, *94*, 106416.

Mahmoodjanloo, M., Tavakkoli-Moghaddam, R., Baboli, A., & Bozorgi-Amiri, A. (2022). Distributed job-shop rescheduling problem considering reconfigurability of machines: A self-adaptive hybrid equilibrium optimiser. *International Journal of Production Research*, *60*(16), 4973–4994.

Massimi, E., Khezri, A., Benderbal, H. H., & Benyoucef, L. (2020). A heuristic-based non-linear mixed integer approach for optimizing modularity and integrability in a sustainable reconfigurable manufacturing environment. *The International Journal of Advanced Manufacturing Technology*, *108*(7), 1997–2020.

Mechaacha, A., Belkaid, F., & Brahimi, N. (2024). Optimisation of process plan generation for reconfigurable manufacturing systems: Efficient heuristics and lower bounds. *International Journal of Production Research*, *0*(0).

Mohapatra, P., Benyoucef, L., & Tiwari, M. (2013). Realising process planning and scheduling integration through adaptive setup planning. *International Journal of Production Research*, *51*(8), 2301–2323.

Musharavati, F., & Hamouda, A. (2012). Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Advances in Engineering Software*, *45*(1), 80–90.

Musharavati, F., Ismail, N., Hamouda, A. M. S., & Ramli, A. R. (2008). A metaheuristic approach to manufacturing process planning in reconfigurable manufacturing systems. *Jurnal Teknologi*, 55â–70.

Nahavandi, S. (2019). Industry 5.0a human-centric solution. *Sustainability*, *11*(16), 4371.

Okhrin, I., & Richter, K. (2011a). The linear dynamic lot size problem with minimum order quantity. *International Journal of Production Economics*, *133*(2), 688–693.

Okhrin, I., & Richter, K. (2011b). An o (t3) algorithm for the capacitated lot sizing problem with minimum order quantities. *European Journal of Operational Research*, *211*(3), 507–514.

Ou, J., & Feng, J. (2019). Production lot-sizing with dynamic capacity adjustment. *European Journal of Operational Research*, *272*(1), 261–269.

Park, Y. W., & Klabjan, D. (2015). Lot sizing with minimum order quantity. *Discrete Applied Mathematics*, *181*, 235–254.

Phadke, M. S. (1995). *Quality engineering using robust design*. Prentice Hall PTR.

Phanden, R. K., Jain, A., & Verma, R. (2011). Integration of process planning and scheduling: A state-of-the-art review. *International Journal of Computer Integrated Manufacturing*, *24*(6), 517–534.

Pickardt, C. W., & Branke, J. (2012). Setup-oriented dispatching rules–a survey. *International Journal of Production Research*, *50*(20), 5823–5842.

Qin, J., Liu, Y., & Grosvenor, R. (2016). A categorical framework of manufacturing for industry 4.0 and beyond. *Procedia cirp*, *52*, 173–178.

Quadt, D., & Kuhn, H. (2008). Capacitated lot-sizing with extensions: A review. *4OR*, *6*(1), 61–83.

Rembold, U., Nnaji, B. O., & Storr, A. (1993). *Computer integrated manufacturing and engineering*. Addison-Wesley Longman Publishing Co., Inc.

Rohaninejad, M., Hanzálek, Z., Tavakkoli-Moghaddam, R., & Vahedi-Nouri, B. (2022). Integrated lot-sizing and job shop scheduling benefiting from reconfigurable machine tools. *IFAC-PapersOnLine*, *55*(10), 1284–1289.

Rohaninejad, M., Vahedi-Nouri, B., Hanzálek, Z., & Tavakkoli-Moghaddam, R. (2024). An integrated lot-sizing and scheduling problem in a reconfigurable manufacturing system under workforce constraints. *International Journal of Production Research*, *62*(11), 3994–4013.

Rong, Y. K., & Huang, S. (2005). *Advanced computer-aided fixture design*. Elsevier.

Roshani, A., Paolucci, M., Giglio, D., Demartini, M., Tonelli, F., & Dulebenets, M. A. (2023). The capacitated lot-sizing and energy efficient single machine scheduling problem with sequence dependent setup times and costs in a closed-loop supply chain network. *Annals of Operations Research*, *321*(1), 469–505.

Rothaermel, F. T., Hitt, M. A., & Jobe, L. A. (2006). Balancing vertical integration and strategic outsourcing: Effects on product portfolio, product success, and firm performance. *Strategic management journal*, *27*(11), 1033–1056.

Sabioni, R. C., Daaboul, J., & Le Duigou, J. (2021). An integrated approach to optimize the configuration of mass-customized products and reconfigurable manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *115*(1), 141–163.

Sadaiah, M., Yadav, D., Mohanram, P., & Radhakrishnan, P. (2002). A generative computer-aided process planning system for prismatic components. *The International Journal of Advanced Manufacturing Technology*, *20*, 709–719.

Scallan, P. (2003). *Process planning: The design/manufacture interface*. Elsevier.

Shabaka, A., & ElMaraghy, H. A. (2008). A model for generating optimal process plans in rms. *International Journal of Computer Integrated Manufacturing*, *21*(2), 180–194.

Shang, X., Milisavljevic-Syed, J., Huang, S., Wang, G., Allen, J. K., & Mistree, F. (2021). A key feature-based method for the configuration design of a reconfigurable inspection system. *International Journal of Production Research*, *59*(9), 2611–2623.

Shin, K. S., Park, J.-O., & Kim, Y. K. (2011). Multi-objective fms process planning with various flexibilities using a symbiotic evolutionary algorithm. *Computers & Operations Research*, *38*(3), 702–712.

Skärin, F., Rösiö, C., & Andersen, A.-L. (2022). Considering sustainability in reconfigurable manufacturing systems research–a literature review. *SPS2022*, 781–792.

Sormaz, D., & Khoshnevis, B. (1997). Process planning knowledge representation using an object-oriented data model. *International Journal of Computer Integrated Manufacturing*, *10*(1-4), 92–104.

Stief, P., Burgat, G., Pour-Massahian-Tafti, M., Dantan, J.-Y., & Siadat, A. (2023). A pragmatic optimization-based approach for analysis and configuration of a reconfigurable multi-product assembly line in the automotive industry. *The International Journal of Advanced Manufacturing Technology*, *129*(9), 3993–4010.

Swamidass, P. M. (2000). *Encyclopedia of production and manufacturing management*. Springer Science & Business Media.

Taha, H. A., & Taha, H. A. (2003). *Operations research: An introduction* (Vol. 7). Prentice hall Upper Saddle River, NJ.

Taillard, É. D. (2023). *Design of heuristic algorithms for hard optimization: With python codes for the travelling salesman problem*. Springer Nature.

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. John Wiley & Sons.

Toczylowski, E. (1995). An o (t/sup 2/) algorithm for the lot-sizing problem with limited inventory levels. *Proceedings 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation. ETFA'95*, *3*, 77–85.

Toussaint, J., & Cheng, K. (2002). Design agility and manufacturing responsiveness on the web. *Integrated Manufacturing Systems*, *13*(5), 328–339.

Touzout, F. A., & Benyoucef, L. (2018). Sustainable multi-unit process plan generation in a reconfigurable manufacturing environment: A comparative study of three hybrid-meta-heuristics. *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, *1*, 661–668.

Touzout, F. A., & Benyoucef, L. (2019a). Multi-objective multi-unit process plan generation in a reconfigurable manufacturing environment: A comparative study of three hybrid metaheuristics. *International Journal of Production Research*, *57*(24), 7520–7535.

Touzout, F. A., & Benyoucef, L. (2019b). Multi-objective sustainable process plan generation in a reconfigurable manufacturing environment: Exact and adapted evolutionary approaches. *International Journal of Production Research*, *57*(8), 2531–2547.

Touzout, F. A., Benyoucef, L., Benderbal, H. H., & Dahane, M. (2018). A hybrid multi-objective based approach for sustainable process plan generation in a reconfigurable manufacturing environment. *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 343–348.

Vahedi-Nouri, B., Tavakkoli-Moghaddam, R., Hanzálek, Z., & Dolgui, A. (2022). Workforce planning and production scheduling in a reconfigurable manufacturing system facing the covid-19 pandemic. *Journal of Manufacturing Systems*, *63*, 563–574.

Vahidinasab, V., & Jadid, S. (2010). Normal boundary intersection method for suppliers strategic bidding in electricity markets: An environmental/economic approach. *Energy Conversion and Management*, *51*(6), 1111–1119.

Van Vyve, M. (2007). Algorithms for single-item lot-sizing problems with constant batch size. *Mathematics of Operations Research*, *32*(3), 594–613.

Vickery, S. K., & Markland, R. E. (1986). Multi-stage lot sizing in a serial production system. *International Journal of Production Research*, *24*(3), 517–534.

Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, *5*(1), 89–96.

Wiendahl, H.-P., ElMaraghy, H. A., Nyhuis, P., Zäh, M. F., Wiendahl, H.-H., Duffie, N., & Brieke, M. (2007). Changeable manufacturing-classification, design and operation. *CIRP annals*, *56*(2), 783–809.

Wilcoxon, F. (1947). Probability tables for individual comparisons by ranking methods. *Biometrics*, *3*(3), 119–122.

Wolsey, L. A. (1989). Uncapacitated lot-sizing problems with start-up costs. *Operations Research*, *37*(5), 741–747.

Xu, X., Lu, Y., Vogel-Heuser, B., & Wang, L. (2021). Industry 4.0 and industry 5.0inception, conception and perception. *Journal of Manufacturing Systems*, *61*, 530–535.

Xu, X., Wang, L., & Newman, S. T. (2011). Computer-aided process planning–a critical review of recent developments and future trends. *International Journal of Computer Integrated Manufacturing*, *24*(1), 1–31.

Yazdani, M. A., Khezri, A., & Benyoucef, L. (2022). Process and production planning for sustainable reconfigurable manufacturing systems (srmss): Multi-objective exact and heuristic-based approaches. *The International Journal of Advanced Manufacturing Technology*, *119*(7), 4519–4540.

Yelles-Chaouche, A. R., Gurevsky, E., Brahimi, N., & Dolgui, A. (2021). Reconfigurable manufacturing systems from an optimisation perspective: A focused review of literature. *International Journal of Production Research*, *59*(21), 6400–6418.

Youssef, A. M., & ElMaraghy, H. A. (2006). Assessment of manufacturing systems reconfiguration smoothness. *The International Journal of Advanced Manufacturing Technology*, *30*, 174–193.

Zangwill, W. I. (1966). A deterministic multi-period production scheduling model with backlogging. *Management Science*, *13*(1), 105–119.

Zhang, Q., & Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, *11*(6), 712–731.

Zhang, S., & Wong, T. (2016). Studying the impact of sequence-dependent set-up times in integrated process planning and scheduling with e-aco heuristic. *International Journal of Production Research*, *54*(16), 4815–4838.

Zhang, Y., Saravanan, A., & Fuh, J. (2003). Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research*, *41*(3), 611–628.

Abstract .The dynamic nature of modern markets has rendered it imperative for manufacturing enterprises to evolve their practices in order to remain competitive. This has led to the development of reconfigurable manufacturing systems (RMS), which offer a solution to the aforementioned challenge. RMS is capable of adapting to fluctuations in demand in terms of both quantity and product type. This makes classical design and planning methods obsolete and necessitates the development of new tools that can effectively facilitate the implementation of RMS. This thesis focuses on process and production planning. The following problems are discussed: (i) process plan generation in a mono-objective context; (ii) process planning integrated with product sequencing in a multi-objective context; and (iii) lot sizing with a focus on exact solution approaches. All proposed approaches, whether exact or approximate, have demonstrated good solution quality through numerical experiments.
Keywords Production planning, Process planning, Reconfigurable manufacturing system, Lot-sizing, Mathematical Modeling, Metaheuristics
الملخص لقد جعلت الطبيعة الديناميكية للأسواق الحديثة من الضروري لمؤسسات التصنيع تطوير ممارساتها من أجل الحفاظ على قدرتها التنافسية. وقد أدى ذلك إلى تطوير أنظمة التصنيع القابلة لإعادة الترتيب (RMS)، والتي تقدم حلاً للتحدي المذكور أعلاه. إن أنظمة إدارة التصنيع القابلة لإعادة الترتيب قادرة على التكيف مع التقلبات في الطلب من حيث الكمية ونوع المنتج. وهذا يجعل أساليب التصميم والتخطيط الكلاسيكية غير مجدية ويستلزم تطوير أدوات جديدة قادرة على تسهيل استخدام نظام إدارة النظم القابلة لاعادة الترتيب بفعالية. تركز هذه الأطروحة على تخطيط العمليات والإنتاج، وتقترح أساليب فعالة لإنشاء خطط العمليات وخطط الإنتاج. تم مناقشة المشكلات التالية: (1) توليد خطط العمليات في سياق أحادي الهدف؛ (2) تخطيط العمليات المتكامل مع تسلسل المنتجات في سياق متعدد الأهداف؛ (3) تحديد حجم الدفعات مع التركيز على أساليب الحل الدقيقة. وقد أظهرت جميع النُهج المقترحة، سواء كانت دقيقة أو تقريبية، جودة حل جيدة من خلال التجارب العددية.
الكلمات المفتاحية تخطيط الإنتاج، تخطيط العمليات، نظام التصنيع القابل لإعادة الترتيب، تخطيط حجم الدفعات، النمذجة الرياضية، خوارزميات الاستدلالات الفوقية.
Résumé En raison de la nature dynamique des marchés modernes, les entreprises manufacturières doivent impérativement évoluer dans leurs pratiques afin de maintenir leur compétitivité. Cela a conduit au développement de systèmes de production reconfigurables (RMS), qui offrent une solution à ces défis. Les systèmes de production reconfigurables sont capables de s'adapter aux fluctuations de la demande en termes de quantité et de type de produit. Cela rend les méthodes classiques de conception et de planification obsolètes et nécessite le développement de nouveaux outils qui peuvent faciliter efficacement la mise en uvre des RMS. Cette thèse se concentre sur la planification des processus et de la production. Les problèmes suivants sont abordés : (i) la génération de plans de processus dans un contexte mono-objectif ; (ii) la planification de processus intégrée au séquencement des produits dans un contexte multi-objectif ; et (iii) le dimensionnement des lots, en mettant l'accent sur les approches de résolution exactes. Toutes les approches proposées, qu'elles soient exactes ou approximatives, ont démontré une bonne qualité de solution par des expériences numériques.
Mots-clés Planification de la production, Planification de processus, Système de production reconfigurable, Dimensionnement des lots, Modélisation mathématique, Métaheuristique.