



**HAL**  
open science

# Concurrent optimization of modular products and reconfigurable manufacturing systems configuration: a customer-oriented offer for mass customization

Rachel Campos Sabioni

► **To cite this version:**

Rachel Campos Sabioni. Concurrent optimization of modular products and reconfigurable manufacturing systems configuration: a customer-oriented offer for mass customization. Other. Université de Technologie de Compiègne, 2021. English. NNT: 2021COMP2613 . tel-03599154

**HAL Id: tel-03599154**

**<https://theses.hal.science/tel-03599154v1>**

Submitted on 6 Mar 2022

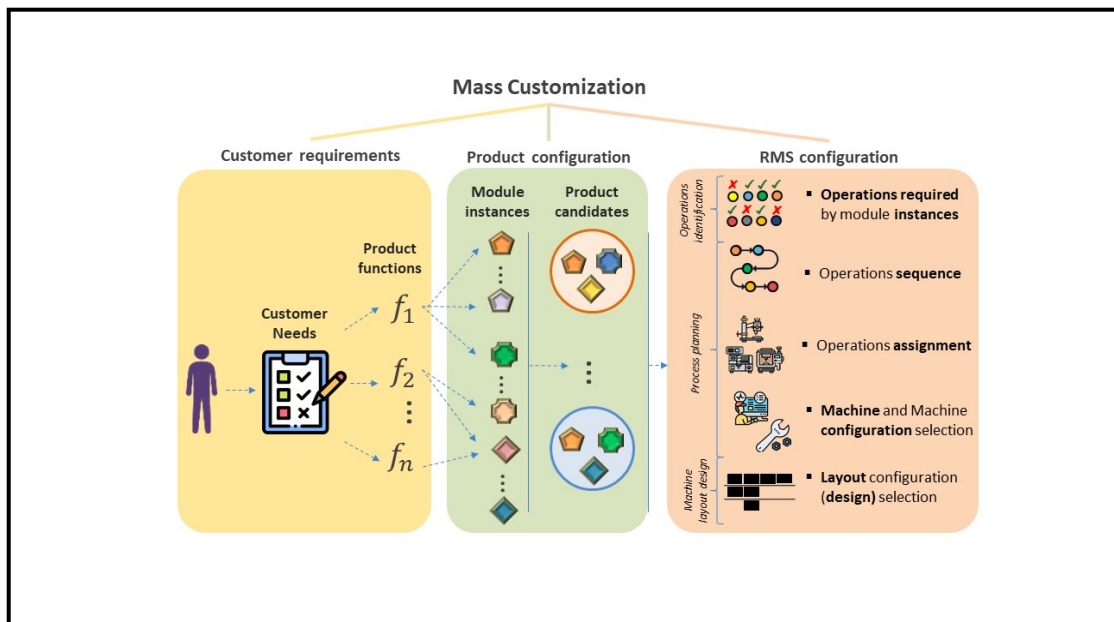
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par Rachel CAMPOS SABIONI

*Concurrent optimization of modular products and reconfigurable manufacturing systems configuration: a customer-oriented offer for mass customization*

Thèse présentée  
pour l'obtention du grade  
de Docteur de l'UTC



Soutenue le 17 juin 2021

**Spécialité :** Génie Industriel : Unité de recherche en Mécanique -  
Laboratoire Roberval (FRE UTC - CNRS 2012)

D2613

Thèse présentée par Rachel CAMPOS SABIONI

Pour l'obtention du grade de Docteur de l'Université de Technologie de Compiègne

Spécialité : Génie Industriel

**Concurrent optimization of modular products and reconfigurable manufacturing systems configuration: a customer-oriented offer for mass customization**

Membres du jury:

<i>Président du Jury:</i>	Pr. Marc ZOLGHADRI	Professeur, Institut Supérieur de Mécanique de Paris
<i>Rapporteurs:</i>	Pr. Catherine DA CUNHA Pr. Farouk YALAOUI	Professeur, Ecole Centrale de Nantes Professeur, Université de technologie de Troyes
<i>Examineurs:</i>	Pr. Antoine JOUGLET Dr. Joanna DAABOUL Dr. Julien LE DUIGOU	Professeur, Université de technologie de Compiègne Maître de Conférences, Université de technologie de Compiègne Enseignant Chercheur, Université de technologie de Compiègne

Université de Technologie de Compiègne

17 juin 2021



*“The world is in the hands of those who have the courage to dream and risk living their dreams”*

*Paulo Coelho*

## ACKNOWLEDGEMENTS

First, I would like to thank God for guiding my paths and blessing my thoughts.

I would like to express my enormous gratitude to my parents, for their unconditional love, support, and valuable life lessons. I would not have gotten this far without both of you. I extend this gratitude to my brother for always being there for me.

I would like to give special thanks to Youssef, for all his love, encouragement and for making life easier and more enjoyable during the hard times.

So many people were a part of my life during these three years of PhD thesis, who helped me somehow and with whom I shared several good moments for which I am very grateful. Family and friends from Brazil and those I met in France: thank you very much!

I would like to thank my advisors, Joanna Daaboul and Julien Le Duigou, for their guidance throughout this thesis and for having pushed me to always do better. They were always willing to share their knowledge and discuss new ideas.

I would also like to thank Professors Catherine da Cunha and Farouk Yalaoui for accepting to reviewer my work and giving me useful feedback.

Also, many thanks to Professors Antoine Jouglet and Marc Zolghadri, members of the jury, for all their relevant questions and comments.

Last, but not least, thanks to all professors and colleagues from the UTC, especially SIPP team and the PhD students and colleagues from the IM building. Thank you all for all the help, French teaching and good moments shared.

## ABSTRACT

Reconfigurable Manufacturing Systems (RMS) emerged from companies' needs to increase their responsiveness to an uncertain market, in which customers are increasingly demanding mass-customised products. Companies focused on mass customisation mainly use the modular product design strategy to cost-effectively provide a large product variety. Hence, coupling the modular product design with the manufacturing in RMS seems to be a good strategy to effectively provide mass-customised products with lower costs.

This work proposes a 0-1 nonlinear integer programming model to jointly optimize the modular product configuration and RMS configuration, which includes the process planning and layout design, for minimizing the total cost of manufacturing products driven by specific customer requirements in mass customization contexts. The costs include raw material, operations, material handling, and reconfiguration of machines and layout. The mathematical model is validated through an exact approach. The analysis carried out confirms the RMS configuration impact on the optimal product choice and its total cost. Two solution approaches based on an evolutionary algorithm are proposed and validated with the exact one. Two case studies are suggested to illustrate the ability of proposed approaches to minimize the total cost of concurrently selecting the product and RMS configurations for mid-size problems with relatively low computation time while ensuring all customer requirements satisfaction.

## RESUME

Les systèmes de fabrication reconfigurables (SFR) sont nés du besoin des entreprises d'augmenter leur réactivité face à un marché incertain, caractérisé par une demande croissante de produits customisés en masse. Les entreprises axées sur la customisation de masse utilisent principalement la conception de produits modulaires pour fournir une grande variété de produits, tout en restant rentable. Par conséquent, coupler la conception de produits modulaires avec sa fabrication en SFR semble être une bonne stratégie pour fournir des produits customisés en masse à moindre coût.

Ce travail propose un modèle de programmation à nombres entiers non-linéaire pour optimiser simultanément la configuration d'un produit modulaire et du SFR, qui comprend la planification de processus et la conception de l'atelier de production, afin de minimiser le coût total de fabrication du produit selon des exigences spécifiques du client. Les coûts comprennent les matières premières, les opérations, le transfert entre machines et la reconfiguration des machines et de l'atelier. Le modèle mathématique est validé par une approche exacte. L'analyse réalisée confirme l'impact de la configuration du SFR sur le choix optimal du produit et son coût total. Deux approches de solution basées sur un algorithme évolutionnaire sont proposées et validées avec l'approche exacte. Deux études de cas illustrent la capacité des approches proposées à minimiser le coût total de la sélection simultanée de la configuration du produit et celle du SFR pour les problèmes de taille moyenne à un temps de calcul relativement faible tout en garantissant la satisfaction de toutes les exigences du client.



## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	IV
ABSTRACT .....	V
RESUME.....	VI
TABLE OF CONTENTS .....	VII
LIST OF TABLES .....	IX
LIST OF FIGURES .....	XI
GLOSSARY .....	XIII
INTRODUCTION.....	1
CHAPTER 1- LITERATURE REVIEW .....	8
1.1. Mass customization.....	8
1.2. Modularity.....	11
1.3. Modular Product Design strategy .....	13
1.3.1. Module definition .....	13
1.3.2. Module interface analysis.....	14
1.3.3. Modular Product Architecture.....	15
1.3.3.1. Product family architecture .....	16
1.3.4. Product configuration .....	18
1.3.4.1. Product configurators.....	24
1.4. Reconfigurable Manufacturing Systems .....	26
1.4.1. Design of Reconfigurable Manufacturing Systems .....	29
1.4.2. Configuration selection in Reconfigurable Manufacturing Systems.....	30
1.4.2.1. Overview of papers found in literature .....	35
1.4.2.2. The RMS configuration-levels .....	36
1.4.2.3. Modelling and optimizing the RMS configuration.....	39
1.5. Integrating configuration decisions from modular product and process .....	41
1.5.1. Integrating configuration decisions of modular product and RMS .....	43
1.5.2. Overview of papers found in literature .....	44
1.5.3. Concurrent optimization problems: modelling and solving .....	46
1.6. Final considerations from this chapter .....	50
CHAPTER 2 - CONCEPTUAL AND MATHEMATICAL MODELS.....	54
2.1. Explaining the main assumptions .....	56
2.1.1. Individual customer requirements .....	56
2.1.2. Operations identification .....	57

2.1.3.	Machine configuration.....	57
2.1.4.	Process planning .....	57
2.1.5.	Layout configuration .....	57
2.2.	Mathematical model propositions .....	58
2.2.1.	Proposition 1 .....	61
2.2.1.1.	ILP model.....	61
2.2.2.	Proposition 2 .....	67
2.2.2.1.	NLIP model.....	68
2.3.	Final considerations from this chapter.....	75
CHAPTER 3 - SOLUTION APPROACHES.....		78
3.1.	Solution approaches proposition .....	78
3.1.1.	Genetic algorithm .....	79
3.1.2.	GA-based solution approaches .....	81
3.1.2.1.	Chromosome encoding and decoding and constraints handling .....	82
3.1.2.2.	Exhaustive search + GA .....	88
3.1.2.3.	Messy GA.....	91
3.2.	Final considerations from this chapter .....	97
CHAPTER 4 – PROPOSITIONS’ VALIDATION .....		99
4.1.	Methods to validate the mathematical models .....	99
4.2.	Mathematical models validation.....	102
4.3.	Evaluation of the initial machines’ configurations changeovers.....	104
4.4.	Resolution approaches validation.....	109
4.4.1.	GA parameters tuning.....	109
4.4.2.	Applying optimal GA parameters.....	113
4.4.2.1.	Solving the ILP problem (Proposition 1) .....	113
4.4.2.2.	Solving the NLIP problem (Proposition 2).....	115
4.5.	Final considerations from this chapter.....	117
CHAPTER 5 - CASE STUDIES .....		119
5.1.	Case 1: Modular smartphone .....	119
5.2.	Case 2: Modular sneaker .....	128
5.3.	Analysis and discussion.....	135
GENERAL CONCLUSION.....		139
BIBLIOGRAPHY .....		143
PUBLICATIONS .....		155
APPENDICES.....		156

## LIST OF TABLES

Table 1. Benefits of product modularity. ....	12
Table 2 Summary of papers addressing product configuration solving problems in mass customization contexts. ....	23
Table 3. Comparison between the three main manufacturing systems' types. Adapted from Koren and Shpitalni (2010) and Koren (2010). ....	28
Table 4. Description of the three main RMS configuration levels and their sub-categories in terms of layout design. ....	32
Table 5 Summary of papers found in literature addressing RMS configuration optimization. ....	33
Table 6 Approaches used to model and solve optimization problems of RMS configuration. ....	40
Table 7. Papers selected from the literature review. ....	45
Table 8. Papers optimizing the configuration of modular products and RMS concurrently. ..	49
Table 9. Set of subproblems addressed by each of the mathematical models proposed in this work. ....	83
Table 10. Input parameters used in the example to validate both mathematical models. ....	102
Table 11. Results obtained with Exhaustive search + CPLEX for both mathematical problems: 1) ILP and 2) NLIP. ....	103
Table 12. Results obtained by Exhaustive search-CPLEX on solving the NLIP problem, with displacement costs equal to 1) 40%, 2) 10% and 3) 5% of the initial displacement cost. ....	104
Table 13. Fragment of the list of all possible initial machines' configurations. ....	105
Table 14. Product variant candidates for each group of selected functions and the frequency they appeared as an optimal solution within the 120 runs. ....	106
Table 15. Summary of the results obtained by each group of functions considering different initial machines' configurations. ....	107
Table 16. Evaluated GA parameters and their tested levels for ESA-GA and mGA methods. ....	109
Table 17. Functions and their adjusted $R^2$ . ....	110
Table 18. Optimal GA parameter values according to the method and problem type. ....	113
Table 19. Results obtained for the <i>ILP</i> problem. ....	114
Table 20. Results obtained for the <i>NLIP</i> problem. ....	116
Table 21. Modules and module instances of the modular smartphone. ....	120
Table 22. Module instances capable of satisfying each available functional requirement. ...	121

Table 23. Input parameters values.....	122
Table 24. Evaluated GA parameters and their correspondent tested levels for smartphone’s case study.....	122
Table 25. Functions modelled and their R <sup>2</sup> adjusted for mGA and ESA-GA.....	123
Table 26. Optimal GA parameter values.....	123
Table 27. Results obtained for the modular smartphone.....	124
Table 28. Anderson Darling and 2 Sample t-tests for the distributions of total cost obtained by ESA-GA and mGA for the smartphone case study. ....	126
Table 29. Available modules and correspondent module instances of sneakers. ....	128
Table 30. Module instances capable of satisfying each available functional requirement. ...	129
Table 31. Evaluated GA parameters and their correspondent tested levels for sneakers’ case study.....	130
Table 32. Functions modelled and their R <sup>2</sup> adjusted for mGA and ESA-GA.....	130
Table 33. Optimal GA parameter values for the sneaker’s example.....	131
Table 34. Input parameters values.....	131
Table 35. Results obtained for a modular customizable sneaker. ....	132
Table 36. Anderson Darling and 2 Sample t tests for the distributions of total cost obtained by ESA-GA and mGA. ....	134

## LIST OF FIGURES

Figure 1. Summary of all chapters presented in this work.....	7
Figure 2. The House of Mass Customization (Paes et al. 2018). .....	9
Figure 3. Relation between product, product modules, module instances and product variants. Adapted from Xu and Liang (2005).....	14
Figure 4. Examples of coupled (integral products) and de-coupled (modular products) interfaces between the whole box-bed (Ulrich 1995). .....	15
Figure 5. Overall view of the modular product development process from the customer requirements to the process design. Adapted from Jiao et al. (2007). .....	18
Figure 6. The product configuration process, which corresponds to the selection of modules that will compose the product variants. The module selection is constrained by some specifications (physical, customer requirements, etc.). .....	20
Figure 7. Illustration of a) Reconfigurable Machine Tool with two different configurations and b) Reconfigurable Inspection Machine (Koren 2010).....	28
Figure 8. Impact of coupling modular product design strategy to the RMS on mass customization.....	51
Figure 9. Illustration of layout reconfiguration. ....	58
Figure 10. Conceptual model of the concurrent optimization of modular product and RMS configurations, driven by individual customer requirements.....	60
Figure 11. GA encoding. ....	83
Figure 12. Decoding the chromosome: a) Part 1: module instance selection and b) Part 2 and 3: operations sequence and selection of machine-configuration for each operation and c) Part 4: selection of layout locations for each machine.....	84
Figure 13. Standard one-point crossover for real-coded strings (left) and heuristic one-point crossover for integer-coded values. ....	86
Figure 14. Illustration of the random float and swap mutations. ....	87
Figure 15. The framework of the ESA+GA approach for solving the ILP (Proposition 1) and NLIP (Proposition 2) problems.....	90
Figure 16. The framework of the pre-processing method applied to get good individuals to inject into the initial population of mGA. ....	93
Figure 17. The framework of the messy GA.....	96
Figure 18. Framework of the exhaustive search and CPLEX for solving the ILP problem. .	100

Figure 19. Framework of the exhaustive search and CPLEX for solving the NLIP problem. .....	101
Figure 20. Main effects plot for the total cost of a) Group 1 b) Group 2, and c) Group 3.....	108
Figure 21. Main effects plot for best fitness for method a) ESA-GA and b) mGA. ....	111
Figure 22. Main effects plot for computation time for method a) ESA-GA and b) mGA. ....	112
Figure 23. Modular smartphone. ....	120
Figure 24 Interval plot of total costs obtained by mGA and ESA-GA in the smartphone case study. ....	125
Figure 25. Illustration of a variant of the customizable modular sneaker. ....	128
Figure 26 Interval plot of total costs obtained by mGA and ESA-GA in the sneaker case study. ....	133
Figure 27. Convergence process of log(Total cost) for mGA (orange line) and ESA-GA (blue line). ....	137

## GLOSSARY

CR: Customer requirement

DML: Dedicated Manufacturing Line

DP: Design Parameter

ESA-CPLEX: Exhaustive Search Algorithm-CPLEX

ESA-GA: Exhaustive Search Algorithm-Genetic algorithm

FMS : Flexible Manufacturing System

FR: Functional requirement

ILP: Integer Linear Programming

MC: Mass customization

MPC: Modular Product Configuration

MPD: Modular product design

mGA: Messy Genetic Algorithm

NLIP: Nonlinear Integer Programming

PDP: Product Development Process

PFA: Product Family Architecture

RAL: Reconfigurable Assembly Line

RAS: Reconfigurable Assembly System

RIM: Reconfigurable Inspection Machine

RMS : Reconfigurable Manufacturing System

RMT: Reconfigurable Machine Tool

# INTRODUCTION

## Research context

Globalization has considerably affected the world markets' behavior, imposing new challenges to the manufacturing industry. The current global market characterized by customers from different countries having different cultures and specific needs has contributed to increasing the demand for customized and personalized products (Koren 2010). From that arises the interest for Mass Customization (MC), a strategy consisting of large product variety production at a cost-effective way to meet customer requirements (Pine 1993a).

Although this strategy has been raised in the 1990s, it has evolved over the years. MC is currently in an advanced stage, in which the focus is no longer on producing a large variety of goods but on meeting each specific customer's requirements (Wang et al. 2017c). Ensuring the desired product at the needed time requires a flexible, agile, and responsive business model mainly focused on the customer and capable of maintaining a close interaction between product development and manufacturing systems (Koren 2010). An appropriate way to keep a close interaction from the customer and product to the process is using modularity strategies.

Reconfigurable Manufacturing Systems (RMS) is the most appropriate modular manufacturing system to succeed in this unpredictable and changeable market. They are systems built around product families composed of machines that can rapidly reconfigure to adjust their capacity and functionality, which increases the RMS's flexibility and responsiveness, enabling them to quickly adapt to new market demands (Koren et al. 1999; Koren and Shpitalni 2010). At the product level, modular product design (MPD) is the best strategy to cope with a wide variety of products. This strategy is based on the design of complex products composed of multiple relatively independent building blocks that can be interchanged in various ways to create different product variants in the same family, enabling a wide product variety from relatively few components (Starr 1965; Ulrich 1994).



Manufacturing highly modular products in RMS seem to be the best strategy for integrating product development with manufacturing systems while enabling the inclusion of individual customer needs in MC contexts. Since both systems have high modularity levels, when coupled, modular products and RMS contribute to increasing overall flexibility, quality control, and the economy of scale while reducing cycle time (Ulrich 1994).

The modularity of products and RMS is strongly related, representing an important issue in the customer and manufacturer relationship, hence a key factor of the successful implementation of MC (Pitiot et al. 2013; Pitiot et al. 2014). A product designed with no consideration of process constraints can imply additional costs and time, directly affecting customer satisfaction. A process configured without consideration of product issues may not fulfil the functionalities necessary to fabricate the product. Therefore both systems must be concurrently analysed to effectively attain modularity benefits while ensuring customer satisfaction (Xu and Liang 2005).

In MC context, the customers are integrated into the value creation process during the product configuration phase. In this phase, customers can choose among many available product feature options the ones they desire in their product, allowing them to get what they want. One drawback of offering extensive options to the customers is that, in many cases, they do not even know what they want, or they do not have enough knowledge about the product to decide which features will be the most appropriate for them.

In such situations, the cognitive complexity can increasingly grow, and customers can experience confusion when facing attractive but excessive options, leading to the “mass confusion” paradigm (Huffman and Kahn 1998a; Chen and Wang 2010; Piller and Walcher 2017). That evidences the importance of helping customers to make their choices during the product customization process.

Although the concurrent optimization of product and RMS configurations driven by individual customer needs is crucial to a successful MC implementation, most research has addressed both issues sequentially instead of concurrently. Further, in many cases, specific customer needs are not even considered when a modular product is

configured to be manufactured in an RMS. Therefore, there is still work to be done on the integration of product and RMS configuration for MC.

This work aims to contribute to this research field by proposing a concurrent optimization of modular products and RMS configurations, driven by specific customer requirements, for mass customizing products.

### **Research challenges**

It is well-known that the MPD is the best strategy to manage a high variety for MC (Feitzinger & Lee, 1997; Jiao & Tseng, 2000). This strategy has been widely investigated for increasing the variety within the product family to produce mass-customized products effectively. In MPD, multiple product variants can be derived from a generic product architecture. This is done at the product configuration phase, in which variants of modules (module instances) are selected to compose products.

In contrast, RMS is a relatively new manufacturing system. They are built around product families, which means that they have enough flexibility to manufacture all parts/products from the same family, considerably reducing costs while keeping high productivity (Koren et al., 1999; Koren, 2010). Their high flexibility is thanks to their ability to reconfigure at machine and layout levels.

While many works have tried to optimize the product configuration according to individual customer requirements in MC contexts, few works address the product configuration integrated with the RMS configuration decisions. The number of studies is still fewer when individual customer needs are considered. The link between product and manufacturing system is stated by the process plan, resulting from the process planning. Process planning or process plan generation is the activity responsible for translating the product design data into the method to fabricate a *single* product, including the machine and configuration specification and operations sequence (Nallakumarasamy et al. 2011; Mohapatra et al. 2013).

When a product is configured according to specific customer requirements, works addressing other manufacturing systems than RMS (systems not reconfigurable) usually embed the production costs into the costs of module instances compounding the

configured product variant. This assumption can be valid for general manufacturing systems because they are not reconfigurable, and then the process plan is only generated once.

However, this may not be an adequate assumption for RMS, because these manufacturing systems can be continuously reconfigured during their life cycle to adapt to new demands coming from the customers. Many machine-configurations can perform a single operation in an RMS, while a single machine-configuration can fulfill different operations. Therefore, a particular product can be manufactured with different costs depending on the operating costs of machine-configurations used to achieve operations as well as on the costs incurred of the required changeovers in machines' configurations.

Besides machines' configurations, the technologies provided by the 4<sup>th</sup> industrial revolution have increasingly enabled companies to redesign/reconfigure their RMS layouts at relatively short time and low costs (Guan et al. 2012). Layout reconfiguration has increasingly become a requirement to adapt the system to produce new product variants in a current unpredictable and changeable market (Maganha et al. 2019). Often addressed as the "*machine layout problem*", the layout configuration is associated with machines' placement into the layout, which greatly impacts material handling costs, lead times, and system productivity (Drira et al. 2007).

Hence, configuring an RMS englobes two main sub-problems: process planning and machine layout configuration. The decision for process planning consists of identifying which pair of machine-configuration will perform each required operation to produce the module instances and in which sequence these operations will be performed. The machine layout configuration, in turn, consists of selecting the best machine placement into the layout.

At the product level, each module instance can satisfy different functions while the same function can be satisfied by various available module instances, but once at a time. Therefore, the configuration decision's focus is selecting the module instances capable of satisfying all customer requirements while ensuring that the final product variant is composed only of compatible module instances.

In such a scenario, many product variants can satisfy a given set of customer requirements at the product level. In contrast, at the process level, there are many process plans and layout design combinations capable of processing a given set of operations required by the module instances selected to compound each product variant candidate. Therefore, this work's main challenge consists of making all these decisions simultaneously, including customer requirements, product and process issues.

Considering the statements previously presented and the challenges of integrating the optimization of product configuration with the RMS configuration (i.e., process planning and layout configuration) for producing mass-customized products to meet specific customer requirements, the main research problem addressed in this work can be summarized by the following question:

*How to integrate the modular product configuration, driven by individual customer requirements, with its process planning and layout configuration in a reconfigurable manufacturing system to minimize the overall manufacturing cost?*

### **Research objective**

This research's main objective is to reduce costs of the customer-driven offer of mass customized products through the concurrent optimization of product configuration, driven by specific customer requirements, and RMS configuration (process planning and layout configuration). The costs include raw material, operations, material handling, and reconfiguration of machines and layout.

### **Main contributions**

Considering the issues previously highlighted, the main contributions of this work are described below:

- I- Proposition and validation of two mathematical models for concurrently optimizing the product configuration, driven by individual customer needs, with the:

- Process planning in an RMS;
- Process planning and layout configuration in an RMS. This second model is an evolution of the first model.

In both cases, the machine's configuration level is addressed, and all customer requirements must be satisfied by the configured product.

- II- The proposition of two solution approaches based on an evolutionary algorithm to ensure the solution of these optimization problems previously presented in high instances at relatively low computation time.

This work aims to contribute to the integrated product and process design (IPPD) field, and more precisely, to the research on the concurrent configuration of product and RMS, which is relatively a new subject of study. It is expected that the concurrent optimization of a product with the process planning and layout configuration in an RMS will contribute to reducing costs of manufacturing mass-customized products in modular and reconfigurable systems.

### **Research structure**

This work is organized in five chapters (Figure 1). The first chapter brings a broad literature review regarding the main subjects addressed in this work, i.e., mass customization, modular product configuration, and RMS configuration, followed by a literature review about works that concurrently optimized the modular product and RMS configurations.

The second chapter presents the conceptual and mathematical models proposed to integrate the optimization of product configuration with the i) process planning only and ii) process planning and layout configuration in an RMS for meeting individual customer requirements in MC contexts.

The third chapter presents both solutions approaches proposed to solve each of the mathematical models presented in chapter two.

The mathematical models' validation is presented in chapter four through the use of a small numerical example.

Chapter five brings two illustrative examples of modular products being processed in RMS to validate the solution approaches' ability to solve problems in larger instances as well as to show how the propositions presented in this work can be applied in industry.

Finally, the conclusion and future research are presented.

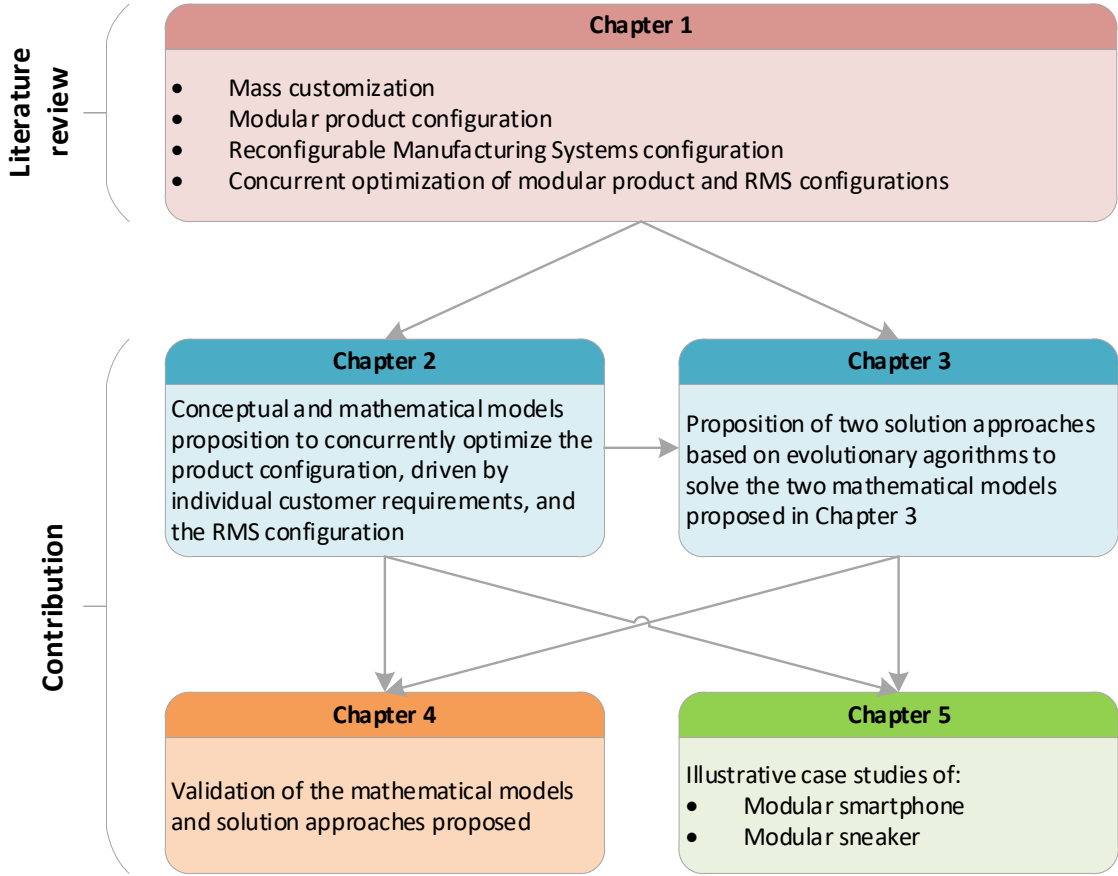


Figure 1. Summary of all chapters presented in this work.

## CHAPTER 1- LITERATURE REVIEW

This thesis addresses the three main following subjects:

1. Mass Customization (MC);
2. Modular Product Configuration (MPC);
3. Reconfigurable Manufacturing Systems (RMS).

The objective of the first part of Chapter 1 is to present a broad review of each of these three concepts separately. Section 1.1 brings the main definitions associated with MC. Then, the modularity concept and its benefits are introduced in Section 1.2. Section 1.3 presents an overview of modular product design, focusing on product configuration, while RMS and issues related to their configuration are presented in Section 1.4. Section 1.5 provides further details about the concurrent optimization of modular products' and RMS configurations.

### 1.1. Mass customization

Mass customization (MC), first raised by Davis (1987) and popularized by Pine (1993a), is a production strategy focused on producing a large variety of customized goods and services, capable of meeting individual customer requirements, at relatively low costs, maintaining mass production efficiency (Pine 1993b; Tseng et al. 2017).

Considering this, one of the main challenges related to the MC strategy is providing products and services individually designed according to each specific customer need through the process' flexibility and agility as well as the integration of product lifecycle (Tseng et al. 2017). MC changes the production paradigm from "made-to-stock" to "made-to-order", in which companies need to provide high product variety in a cost-effective way (Tseng et al. 2017).

According to Feitzinger and Lee (1997) an effective MC program is based on three main organisational-design principles, which are (1) the product (2) the manufacturing process and (3) the supply network, as described below:

1. *Product*: must be designed by many standardized and independent building blocks, which can be easily combined to compose different product variants at low cost;
2. *Manufacturing process*: must be composed by independent production modules that can be rapidly and easily rearranged to support different production requirements;
3. *Supply network*: must be organized in a way that provides enough flexibility to perform cost-effective customization, ensuring a quick response to individual customer orders.

These organizational principles were illustrated by Paes et al. (2018) through the House of MC, in which the first column represents the process, while the second one corresponds to the supply chain management and finally the product comes on the third column. This evidences that a key element of MC is modularity, at product and process level, to ensure enough flexibility and responsiveness to unpredictable market demands.

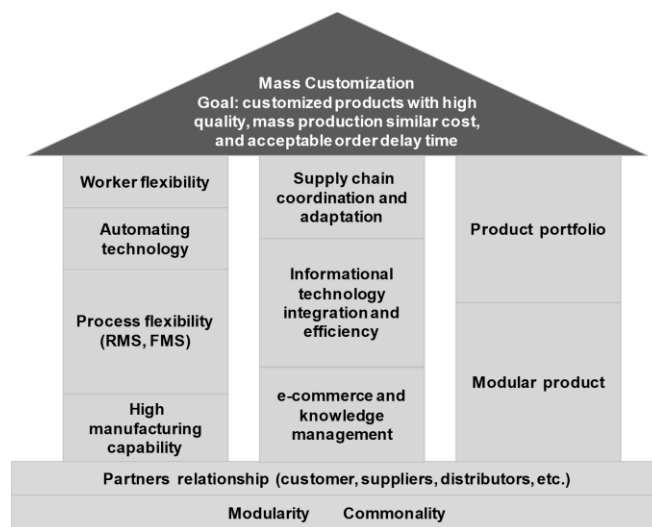


Figure 2. The House of Mass Customization (Paes et al. 2018).

According to Tseng et al. (2017), the main focus of MC is variety and customization through flexibility and responsiveness, which is attained thanks to the production of modular



product families made from modules assembly according to customer needs. This means that customers must be integrated into the product development process to include their individual needs effectively. In this sense, tools allowing customers to co-design or co-configure the product become essential in MC (Franke and Piller 2002; Tseng et al. 2017).

One important tool widely used by companies from different industries, such as footwear, clothing, automobile, etc. is the product configurator system or MC toolkits (Franke and Piller 2002). These knowledge-based systems are the interface between the company and customers that allows the last to customize a given product according to their own needs (Jannach and Zanker 2013; Zheng et al. 2017).

While MC toolkits are a key element of MC, they also represent one of the main sources of complexity of product development, from the customer's point of view (Franke et al. 2004). The lack of customers' knowledge in product features together with the large number of choices available tend to lead customers to the "mass confusion" paradigm, in which they become lost in the wide set of choices and cannot identify what they really want (Piller and Walcher 2017). In other words, MC is not about offering wide and limitless alternatives, instead, it is more about offering a limited but assertive set of options according to real customer needs (Piller and Walcher 2017).

Companies applying the MC strategy usually divide their customers into different market segments, within which they "put" people sharing some characteristics and needs (Wang et al. 2017c). This strategy is effective to get customer needs in a broad view, but it is not able to catch unique needs of individual customers. Each person is unique and people are increasingly demanding for products that meet exactly their individuality and represent their uniqueness (Koren 2010; Piller and Blazek 2014). This customization level aiming to meet each individual customer requirement can be referred to as the advanced stage of MC or mass personalization (Wang et al. 2017c). This type of MC requires products with a high level of modularity, being changeable, reconfigurable and adaptable to meet each individual's specific needs (Wang et al. 2017c).

The production of highly customized or personalized products requires a new business model with a closer interaction between the product development process and the

manufacturing system, and a main focus on the customer, who is the central element (Koren 2010). In other words, decisions related to product development must integrate manufacturing system decisions as well as customer requirements to ensure the production of high-quality products that meet customer expectations.

According to Wang et al. (2017c), the new technological advances provided by the 4<sup>th</sup> industrial revolution, such as internet of things, cyber-physical systems, etc. have contributed to increase the manufacturing systems' responsiveness and are some of the main enablers of MC at high level or mass personalization. The next sub-sections (1.2, 1.3 and 1.4) will present more details concerning modularity at product and process level and how it acts as a main MC enabler.

## **1.2. Modularity**

The concept of modularity began to emerge in the sixties with Starr (1965), but it is only in the nineties that it started to stand out in scientific literature with the publication of Ulrich (1994). According to both authors, modularity is mainly related to the design of a complex product or process composed by many relatively independent elements, also called building blocks, which can be interchanged in several ways to create different product variants.

Modularity is also related to the coupling degree between building blocks (Schilling 2000). The higher the ability of a product or a process to disaggregate and recombine into new configurations with low functionality loss, the higher is its modularity degree (Schilling and Steensma 2001). Modularity is also related to the minimization of incidental interactions among different modules (Ulrich 1994). The optimal structure of a modular product or process can be achieved by clustering components according to their degree of interaction; it means maximizing interactions inside modules and minimizing interactions outside them (Whitfield et al. 2002).

The main benefits related to product modularity are: (1) Economy of scale and cost reduction, (2) Product variety management, (3) Flexibility, (4) Decoupling of tasks and specialized production, (5) Reduced cycle-time and (6) Quality control and improvement, as

detailed in Table 1. Although these benefits were addressed for product modularity, they can be extended to process modularity.

Table 1. Benefits of product modularity.

<b>Benefit</b>	<b>Description</b>	<b>Authors</b>
<i>Economy of scale and cost reduction</i>	Due to the standardization, the same component can be shared in several varieties of products; consequently, it is produced at high scale, which amortizes capital expenses and development resources related to product variety.	Ulrich (1994); Newcomb et al. (1998); Kusiak (2002); Meng et al. (2007); Jacobs et al. (2007); Shaik et al. (2015)
<i>Product variety</i>	Since modules can be switched among different products, it is possible to produce a large variety of products, from a limited number of modules.	Ulrich (1994); Kusiak (2002); Tu et al. (2004); Hölttä-Otto and de Weck (2007); Jacobs et al. (2007)
<i>Flexibility</i>	Modules can be quickly recombined to adapt to the variations of market demands. Further, considering mass customization, it permits consumers to adapt modules according to their own needs.	Ulrich (1994); Baldwin and Clark (1997); Tu et al. (2004); Voordijk et al. (2006)
<i>Decoupling of tasks and specialized production</i>	Partitioning a complex product in many modules contributes to organizing a complicated product design into several simpler and specialized tasks, that can be accomplished by specialists.	Ulrich (1994); Gershenson et al. (2003)
<i>Reduced cycle-time</i>	Since independent modules are produced separately, they can be manufactured at the same time, which reduces the production cycle-time of consumer demands for customized products. Further, considering that modules can be assembled only at the end of the production line, product modularity contributes to increased productivity, agile manufacturing as well as postponement.	Ulrich (1994); Hoek and Weken (1998); Jacobs et al. (2007); Jacobs et al. (2011); Shaik et al. (2015); Shamsuzzoha et al. (2017)
<i>Quality control and improvement</i>	Since each module has well defined function(s), it is easier to test each one and to isolate potential quality problems. Likewise, it is also simpler to do the product maintenance and repair by separating and exchanging only critical modules.	Ulrich (1994); Feitzinger and Lee (1997); Hoek and Weken (1998); Kusiak (2002); Tu et al. (2004); Jacobs et al. (2007); Shaik et al. (2015)

### **1.3. Modular Product Design strategy**

Modular Product Design is characterized by the development of products composed of many relatively autonomous building blocks that are also called modules (Jiao and Tseng 2000). According to Jiao and Tseng (2000) and Paes et al. (2018), the main issues related to the Modular Product Design are: (1) Module creation/definition; (2) Module interface analysis; (3) Product architecture, and (4) Product configuration. This work is focused only on the product configuration, as previously stated. However, all these issues will be detailed in the following sub-sections because understanding product configuration requires a good comprehension of other Modular Product Design issues.

#### **1.3.1. Module definition**

The module is a component or a set of components that are strongly coupled within and loosely coupled to the rest of the system, being able to be easily removed from the system without destroying it (Newcomb, Bras and Rosen, 1998; Allen and Carlson-Skalak, 1998; Hölttä-Otto and de Weck, 2007; Salvador, 2007). The module definition/identification corresponds to the selection of components that will compose a module which is based on the interactions among them (Shamsuzzoha et al. 2017). In an ideal module, each component has no dependency among all other components that are not contained in the same module (Gershenson et al. 1999).

The module identification can also be made from the functional point of view. In this case, product modules are defined as a set of components interfaced with each other to fulfill a single function in the product (Allen and Carlson-Skalak 1998; Pil and Cohen 2006). According to Ulrich (1994), a product with a completely modular design represents a one-to-one correspondence between each module and functional element.

Once modules are identified/defined, it becomes possible to develop new module instances, which means variants from the same module (Figure 3). For example, in the case of a car, supposing that an engine is defined as a module, the different engine types (diesel, petrol, etc.) are called module instances (Xu and Liang 2005). Increasing the number of module instances

contributes to increasing module instance combinations, and as a consequence, the product variants, which is typically done in MC contexts.

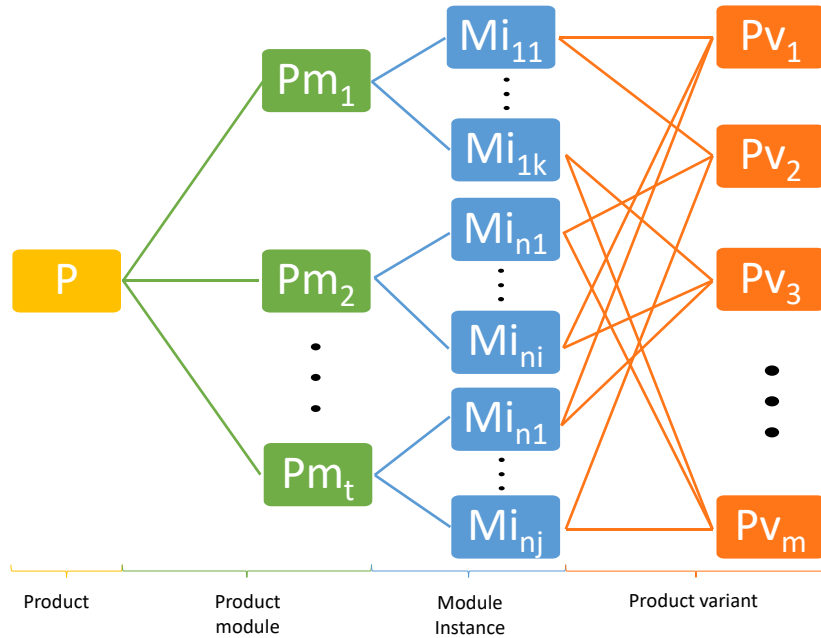


Figure 3. Relation between product, product modules, module instances and product variants. Adapted from Xu and Liang (2005).

### 1.3.2. Module interface analysis

The interface plays an important role in product design since they represent the first interaction between two or more components/modules (Zheng et al., 2016). There are different types of interfaces such as physical/geometric, energetic, informational, and material (Pimmler & Eppinger, 1994; Ulrich, 1995; Mikkola & Gassmann, 2003; Zheng et al., 2016).

The interface between modules is generally de-coupled, meaning that one module's change does not affect another one's functions (Ulrich, 1995; Cabigiosu and Camuffo, 2017). The reversible interface's idea arises from the de-coupled interface concept, which means

that modules are loosely enough coupled between them, being quickly de-coupled and re-organized (Figure 4) (Salvador 2007).

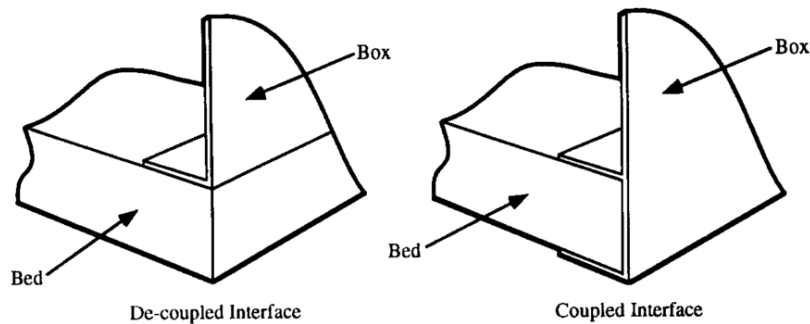


Figure 4. Examples of coupled (integral products) and de-coupled (modular products) interfaces between the whole box-bed (Ulrich 1995).

Since interfaces establish the interactions between components and modules, their standardization and specification degree guide the compatibility between components and modules (Mikkola and Gassmann 2003). Interface standardization means using common and agreed mechanisms/specifications for the interaction between modules, which is essential for modular design (Vickery et al. 2016). Standardized and loosely-coupled interfaces help increasing product variety and production flexibility since the same module can be adopted by several products (Ulrich 1995).

### 1.3.3. Modular Product Architecture

The product architecture represents the scheme in which the functions will be allocated to different modules. Ulrich (1995) and Ulrich and Eppinger (2009) state that the product architecture can be defined into three main steps described as follows.

- I. *Arrangement of functions (functional requirements)*: the set of functional requirements corresponds to the functions that the whole product will fulfill;
- II. *Mapping from functions to modules*: each module will implement the functional requirements in the product. A mapping between functions and modules can be one-to-one (completely modular product), many-to-one, or one-to-many;

III. *Interface specification among different modules*: is the definition of geometric interactions between each physical component (module).

The product architecture is generally established on product development's early steps, impacting other product design steps, such as product configuration (Ulrich 1995). This shows the importance of defining the architecture type and its modularity level to appropriately define the product configuration of new product variants, especially in high product variety scenarios. This topic will be further discussed in Section 1.3.4.

#### **1.3.3.1. Product family architecture**

In MC context, there are some information and knowledge necessary to the product development process (PDP), such as production and process constraints and feasibility, customer's preferences, as well as the information and know-how of similar products previously developed (Daaboul et al. 2011). From this need to maintain the knowledge on PDP arises the concept of Product Family Architecture (PFA), which consists of a generic architecture that catches and uses the commonality, within which several product variants can be derived, by using a common product line structure (Jiao and Tseng 2000).

In addition to the PDP knowledge management, the PFA plays an essential role in the product family design process for MC, offering many product variants, according to individual customer requirements, within a logical framework (Jiao and Tseng 1999). PFAs have two main design characteristics: (1) modularity of the product structure and (2) commonality among product variants (Jiao and Tseng 2000).

Stating a coherent PFA is a prerequisite for designing mass-customized products since it is responsible for synchronizing the traditional activities of product design with sales, marketing, and product service (ElMaraghy et al. 2013). According to Jiao and Tseng (1999), the PFA englobes three perspectives: functional, technical (behavioral), and physical (structural), as described as follows.

*Functional*: consists of product functional requirements/elements (FRs) (Suh 1990; Ulrich 1995), and their interrelationships (Pahl and Beitz 1995).

*Technical or behavioral:* refers to the application of technology by identifying which are the design parameters (or design solutions) (DP) necessary to satisfy the FRs. On modular products, each DP is described in terms of functional features (FF) (Jiao and Tseng 2000).

*Physical or structural:* represents the product information considering its physical design and is strongly related to the product manufacturing (Jiao and Tseng 2000). In other words, the DPs from the technological perspective will be accomplished by many physical modules, components, and subassemblies (Jiao and Tseng 1999). The process constraints highly impact the physical perspective since the objective is to ensure an easy manufacturing and assembly process without affecting the economy of scale (Jiao and Tseng 2000).

These perspectives can be seen as some domains of product family development, in which physical and technical perspectives are coupled into the physical domain, following the customer and functional domains and preceding the manufacturing and logistics (not addressed here) ones, as illustrated in Figure 5 (Jiao et al. 2007).

The customer requirements (CRs), at customer domain, represent the demands of market segments and are translated into functional requirements (FRs) at the functional domain, in which the FRs are structured from a technical point of view, considering engineering issues and available product technologies (Jiao et al., 2007).

The FRs will be translated into design parameters (DPs) in the physical domain, meaning that engineers will choose among available product modules the ones that have all DPs necessary to satisfy the FRs. Linking FRs into DPs allows better technical data management according to the family's functions (Le Duigou et al., 2009). Finally, at the process domain, DPs are mapped into process variables (PVs), which will generate one or multiple candidates of process planning according to the available resources (machines, tools, routings, etc.) (Jiao et al., 2007).



These steps to design the product family show the relevance of considering the product design integrated into the process planning decisions since DPs will lead to the PVs choice. In contrast, an ill-fitted choice of PVs may increase manufacturing costs or even not guarantee appropriate manufacturing of product modules.

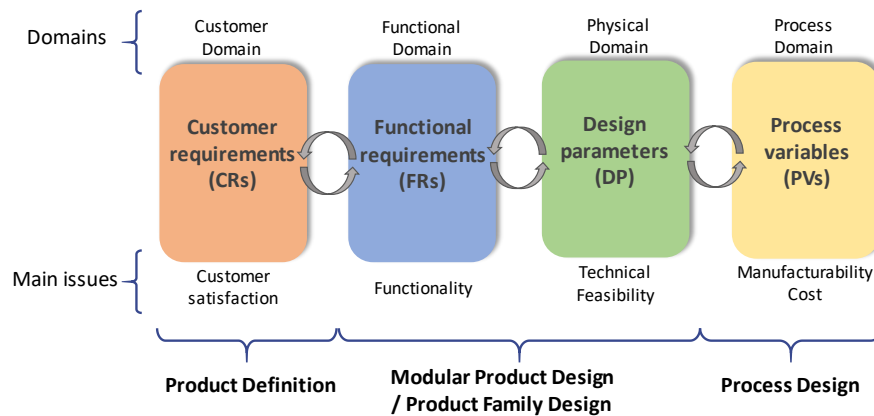


Figure 5. Overall view of the modular product development process from the customer requirements to the process design. Adapted from Jiao et al. (2007).

#### 1.3.4. Product configuration

MC has motivated companies to move from the design of individual products to the development of product families (Sabin and Weigel 1998). As introduced in Section 1.3.3, in MC the focus is on developing a generic product architecture from which multiple product variants from the same family will derive. Once customer needs are correctly understood, a product variant is developed within the previously developed generic product architecture to meet them (Sabin and Weigel 1998). This is done through the product configuration phase.

Product configuration is a process of selecting, from a pre-defined set of elements, which ones will be arranged together to satisfy given specifications (Pahl and Beitz 1995; Sabin and Weigel 1998) (Figure 6). According to Mittal and Frayman (1989) the configuration tasks consists on:

“Given: (A) a fixed, pre-defined set of components, where a component is described by a set of properties, ports for connecting it to other components, constraints at each port that describe the components that can be connected at that port, and other structural constraints (B) some description of the desired configuration; and (C) possibly some criteria for making optimal selections.

Build: One or more configurations that satisfy all the requirements, where a configuration is a set of components and a description of the connections between the components in the set, or, detect inconsistencies in the requirements.”

Considering the configuration of modular products, this is characterized by the arrangement of modules/components according to a generic product structure (product architecture) (Chen and Wang 2010). These modules/components are pre-defined, and then, new components and modules cannot be created, and neither their interface can be modified (Mittal and Frayman 1989; Sabin and Weigel 1998).

Therefore, the product configuration can be summarized as the task of finding feasible solutions (product variants/configurations) from a set of well-defined sub-components/modules that will be combined to satisfy specific requirements while respecting product constraints (e.g., modules interface) (Li et al. 2006; Pitiot et al. 2013; Wang et al. 2017a). Product configuration task can generate different solutions depending on the inputs (requirements). These solutions can include the product’s technical specifications, bill of materials, or even the process plan and costs (Zhang 2014).

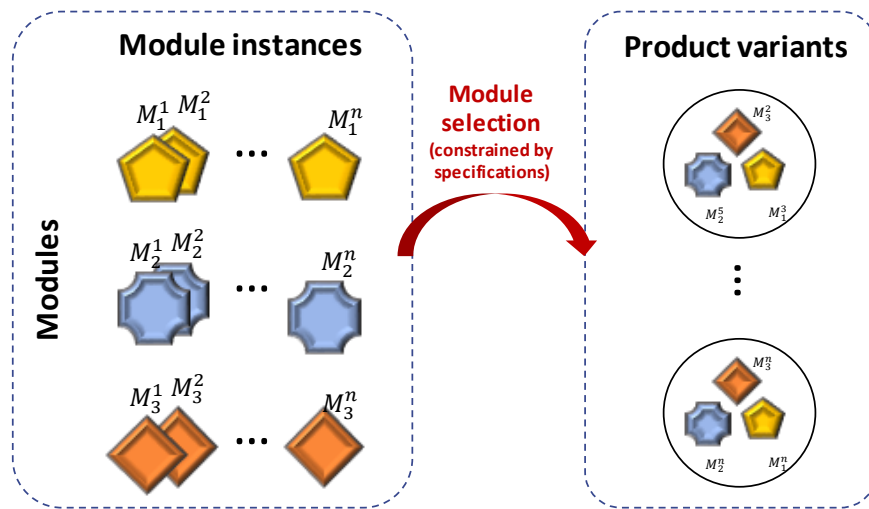


Figure 6. The product configuration process, which corresponds to the selection of modules that will compose the product variants. The module selection is constrained by some specifications (physical, customer requirements, etc.).

In MC contexts, customer requirements can be used as inputs for the modular product configuration task, making that an essential step in mass-customizing products. It allows the combination of multiple standardized modules according to customer requirements, which helps companies increase their product variety at low cost while meeting a wide range of customers' needs (Li et al. 2006; Koren 2010).

One drawback of having flexible and reconfigurable product lines for producing a large variety of products is that it can concern hundreds or thousands of configurable components leading to an increased probability of errors (Sabin and Weigel 1998). These errors can considerably increase production delays and costs, explaining why product configuration is crucial for efficiently mass customizing products (Sabin and Weigel 1998). Besides that, it also evidences how it is important to consider product configuration decisions integrated with process configuration tasks, especially in flexible and reconfigurable manufacturing systems.

According to Zhang (2014), one of the main issues associated with the product configuration is configuration solving, which corresponds to the development and

application of algorithms to solve or optimize configuration tasks. The author identified three types of configuration solving strategies as follows:

- I. *Identification of prime product configurations* based on past sale data analyses by using data mining techniques. This technique is mainly applied when large volumes of sales data are available (Kusiak et al. 2007; Song and Kusiak 2009; Da Cunha et al. 2010; Jiao and Yang 2019);
- II. *Generation of feasible product configurations* according to given customer requirements. Generally formulated as a constraint satisfaction problem (CSP). Works using these strategies are focused on finding feasible product configurations based on the available elements (modules/components) and the constraints established (e.g., components' compatibility, customer requirements, etc.) (Xie et al. 2005; Wang et al. 2011; Yang et al. 2012; Pitiot et al. 2020);
- III. *Optimization of product configuration* is focused on selecting, among all feasible product configurations, the optimal one in terms of given criteria (e.g., cost, customer satisfaction, sustainability factors, etc.) (Li et al. 2006; Hong et al. 2008; Tang et al. 2017; Yang et al. 2020). These problems are also often modeled as CSP when the product configuration is optimized according to given criteria while ensuring that all customer requirements are satisfied (Pitiot et al. 2013; Pitiot et al. 2014). The techniques used to solve optimization problems are generally heuristics, including genetic algorithm, simulated annealing, ant colony optimization, constrained filtering based evolutionary algorithm.

Table 2 presents some papers found in literature addressing product configuration solving problems in MC contexts by using strategies II and III. It means, only papers trying to generate feasible product configurations or optimizing product configuration according to specific customer requirements were considered in Table 2. All of them addressed the individual/specific requirements as constraints.

Using customer requirements (CRs) to constrain CSP and mathematical optimization problems ensure that the final product will fulfill all specific requirements. In some cases, CRs are directly associated to product components/modules while in others the CRs are addressed as functions, which are mapped into product components/modules. The presence of the components/modules on the product means that the CRs or functions were satisfied.

Most of the papers focused on optimizing a given criteria in product configuration tried to minimize costs (Pitiot et al. 2013; Pitiot et al. 2014; Wei et al. 2014; Tong et al. 2015; Pitiot et al. 2020; Yang et al. 2020). Besides them, Pitiot et al. (2013; 2014; 2020) appear to be the only ones that concurrently optimized product configuration and process planning. Although Tong et al. (2015) addressed manufacturing and assembly costs, they considered these costs as constants according to the type of module/component selected.

Besides costs, some papers tried to maximize the product performance (Hong et al. 2008; Wei et al. 2014; Tong et al. 2015) and profit (Yang et al. 2018; Song et al. 2020). Two papers recently published addressed environmental issues through the minimization of greenhouse gas emissions and costs of purchasing carbon emissions (Tang et al. 2017; Yang et al. 2020).

Regarding the approaches to model the configuration problems, it is possible to see from Table 2 that they were mainly modeled as CSP or mathematical programming problems using integer variables; however, some methods based on multi-criteria decision making and AND/OR tree were also found (Hong et al. 2008; Zhu et al. 2008). Papers using CSP approaches were mainly focused on addressing more complex interactions between components through the consideration of structural rules (e.g., mandatory, cardinality, has attribute, etc.) and a higher number of more complex configuration rules (e.g., requires, exclusion, connection type, etc.) (Yang and Dong 2013; Song et al. 2020). In contrast, papers modeling their problems using mathematical programming considered fewer and simpler configuration rules (e.g., compatibility) (Yang et al. 2018; Yang et al. 2020).

When works were focused on finding feasible product configurations without optimizing a given criteria, they were often modeled as a CSP only, being solved by algorithms specific for solving this kind of problems such as backtracking and depth first

search algorithms which are constraint programming (CP) techniques (Wang et al. 2011; Jannach and Zanker 2013; Yang and Dong 2013). However, some papers modelling their problems as CSP used heuristic approaches to optimize a given criteria (Pitiot et al. 2013; 2014; 2020).

Table 2 Summary of papers addressing product configuration solving problems in mass customization contexts.

Author	Optimization objectives		Modelling Approach	Solving Technique
	Min	Max		
Pitiot et al. (2020)	Cost, cycle time	-	CSP	CFB-EA+
Song et al. (2020)	-	Total profit	MILP	CPLEX
Yang et al. (2020)	Cost (of modules and of purchasing carbon emissions)	-	MILP	GA
Yang et al. (2018)	-	Profit	MINLP	Stochastic optimization
Tang et al. (2017)	greenhouse gas (GHG) emissions	Customer satisfaction index	MINLP	GA
Tong et al. (2015)	Cost (manufacturing, assembly, others)	Product performance index	CSP + ILP	Pareto GA
Pitiot et al. (2014)	Cost, cycle time	-	CSP	CFB-EA
Wei et al. (2014)	Cost, time	Performance	MILP	NSGAI+ fuzzy-based select mechanism
Jannach and Zanker (2013)	-	-	GCSP	Asynchronous Backtracking algorithm
Pitiot et al. (2013)	Cost, cycle time	-	CSP	Branch and bound + filtering system/ adapted SPEA2
Yang and Dong (2013)	-	-	CSP	Depth-first search + backtracking search algorithm
Yang and Dong (2012)	Cost	-	MILP	CPLEX
Wang et al. (2011)	-	-	Dynamic CSP	Augmented backtracking method
Kunz et al. (2009)	Total number of components	-	CSP + ILP	Depth-first search + backtracking search algorithm
Hong et al. (2008)	Cost	Performance	Constrained AND/OR tree	Genetic programming
Zhu et al. (2008)	-	Utility value	MCDM	Fuzzy-based method
Li et al. (2006)	Cost, time	-	MINLP	GA
Xie et al. (2005)	-	-	CSP	Search algorithm based on back-jumping

CFB-EA: Constraint Filtering Based – Evolutionary Algorithm; CSP: Constraint Satisfaction Problem; GA: Genetic Algorithm; SPEA: Strength Pareto Evolutionary Algorithm; MCDM: Multi-criteria decision making problem

Works modelling their problems as mixed integer linear models solved them by using CPLEX and GA-based algorithms (Yang and Dong 2012; Wei et al. 2014; Song et al. 2020; Yang et al. 2020). Heuristic techniques, such as GA and stochastic optimization, were used to solve product configuration problems modeled as nonlinear mixed integer programming problems, (MINLP) (Li et al. 2006; Tang et al. 2017; Yang et al. 2018).

In summary, papers addressing configuration-solving problems in MC contexts for meeting specific customer needs always address CRs (or functions) as constraints assuming that they are directly mapped into product components. Therefore, the presence of a given component automatically means that a requirement was satisfied. When product configuration solving problems involve optimization criteria, the main techniques used are heuristics. Few papers have considered process issues when optimizing the product configuration. Section 1.5 will bring more details about works optimizing product module selection or product configuration problems concurrently to process decisions.

#### **1.3.4.1. Product configurators**

Product configuration systems, also known as product configurator or mass customization toolkits (Franke and Piller 2002) are knowledge-based systems responsible for adapting a product according to specific customer needs (Jannach and Zanker 2013; Zheng et al. 2017). They have received increasing interest from academia and industry since Digital Equipment Corporation established the R1 program (after called XCON) to configure VAX (Virtual Address eXtension) computer systems for meeting diverse customer requirements (McDermott 1982).

There are several benefits associated with the use of product configuration systems, such as increasing customer satisfaction and the quality of product specifications as well as the product profitability, but also reducing lead times and routine work (Kristjansdottir et al. 2018). It is no wonder that several companies selling mass-customized products have invested in product configuration systems, such as Dell, Cisco Systems, Rebook, and Nike (Trentin et al. 2012; Piller and Walcher 2017).

In MC, customers can co-design the product with the manufacturer. This strong interaction between customer and manufacturer takes place in the product configuration phase through the product configuration system, which integrates the customers into the value creation by allowing them to configure their products according to their own needs from a set of pre-defined options (Franke and Piller 2002; Chen and Wang 2010). This is why these toolkits are recognized as critical drivers for MC implementation's success or failure (Franke and Piller 2002).

Although product configurators allow customers to contribute into value creation, configuring customized products through these toolkits is one of the main drivers for complexity from customers' perspective due to the knowledge gap between companies and customers (Franke and Piller 2002; Franke et al. 2004). The large set of choices and the unfamiliarity of the customer with the product features can lead to the paradox of choice in mass customization, also known as "mass confusion" (Huffman and Kahn 1998a; Piller and Walcher 2017). Too many options can lead the customer to indecision and, therefore, in many cases, dissatisfaction (Piller and Walcher 2017).

Further, customers usually have no clear idea of what solution (product variant/configuration) might fit their needs. In some cases, they still have to understand their needs (Franke and Piller 2002). Consequently, customers can be uncertain during the product configuration process which potentially leads to their dissatisfaction.

According to Huffman and Kahn (1998a), there are two approaches to present the product varieties in an MC toolkit: (1) attribute-based and (2) alternative-based. In an alternative-based approach, customers are invited to create their product from a set of many product parts (modules) alternatives. While in the attribute-based approach, customers are asked about their product attribute preferences and, based on their answers, a whole product proposition is chosen from a large set of options. Huffman and Kahn (1998a) state that presenting products in terms of their attributes reduces perceived complexity and favours customers' enthusiasm to make their choice and increases their satisfaction. It means that customers prefer not to choose from a long list of customization options but instead express their personal needs (Franke and Piller 2002).



This shows the importance of guiding customers during the product configuration process, helping them find a product option that fulfils their personal requirements. This is why many researchers have been working on the optimization of product configuration driven by individual customer requirements, in which the focus is on getting the customer uniqueness (Hong et al. 2008; Yang et al. 2012; Dou and Zong 2014; Lee et al. 2019). Although these papers have focused on optimizing product configuration in terms of individual customer needs, they did not address process issues (e.g., cost/time of performing operations, changing machine configuration, handling material, etc.) that can affect the final product cost.

MC is currently characterized by an increasing product variety and market unpredictability. To thrive in this scenario, companies have increasingly manufactured their products in flexible or reconfigurable systems, which can vary their production costs or time according to how they are configured and operations are assigned and sequenced. This highlights the importance of optimizing the product configuration not only to meet customer requirements but also to keep the product cost-effective.

The challenges related to product configuration optimization for attaining specific customer requirements while keeping the product cost-effective in terms of manufacturing and assembly processes will be further discussed in Section 1.5.

#### **1.4. Reconfigurable Manufacturing Systems**

According to Koren et al. (1999), there are three main manufacturing systems' types: Dedicated Manufacturing Systems (DMS), Flexible Manufacturing Systems (FMS) and Reconfigurable Manufacturing Systems (RMS). These are compared in Table 3. DMS can use multiple tools simultaneously, being capable of manufacturing high quantity of a single part type at high production rate, keeping a relatively low cost per part when demand is high (Koren et al. 1999). However, when demand is not high enough or bigger than the supply, DMS cannot operate at its full capacity, and then stops being cost-effective (Koren and Shpitalni 2010).

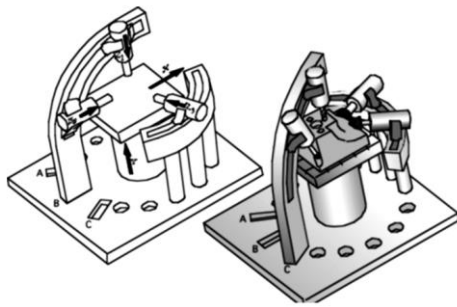
FMS emerged from the need to overcome the limitations of DMS. In contrast to DMSs, FMSs are flexible being capable of producing a variety of products (Koren et al. 1999). This is possible thanks to their computer-numerically-controlled (CNC) machines and other programmable and automation functions (Koren et al. 1999; Koren and Shpitalni 2010). The single-tool operation character of CNC machines makes the throughput of FMS much lower than that of DMS. Further, the high cost of CNC's associated with the low FMS throughput leads to a higher cost per part and to lower production capacity (Koren and Shpitalni 2010).

The current unpredictable and changeable market is characterized by customers from different countries, with different habits and cultures increasingly demanding customized and personalized products. DMS and FMS cannot respond to these changes and demands because of the inflexibility of DMS and the low throughput of FMS.

From the need to respond to these new market tendencies and instability emerged the RMS, which combines the high DMS's throughput with the FMS's flexibility for producing product/part families (Koren et al. 1999). RMS is known as the most appropriate manufacturing system to thrive in this current global market due to its high responsiveness, since it is capable of quickly adjusting its production capacity as well as its functionality to fit new market demands or new regulatory requirements (Koren 2010).

The RMS responsiveness is due to the machines that compose this system. In addition to the CNC, RMS are equipped with the reconfigurable machine tools (RMTs) (Figure 7a) and reconfigurable inspection machines (RIMs) (Figure 7b). RMTs are a new type of modular and changeable machines that allow the system to be rapidly reconfigured to change its capacity or to produce new product varieties (Koren et al. 1999). RIMs are a type of inspection machines that allows non-contact in-line measurement of machined parts in few seconds. Due to that, RIMs are more accurate and faster than computer-controlled part measurement machines (CMMs), an off-line machine inspection that can take up to many hours to complete an inspection of a single part (Koren 2010).

a)



b)

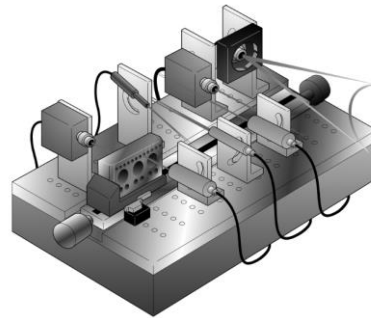


Figure 7. Illustration of a) Reconfigurable Machine Tool with two different configurations and b) Reconfigurable Inspection Machine (Koren 2010).

Table 3. Comparison between the three main manufacturing systems' types. Adapted from Koren and Shpitalni (2010) and Koren (2010).

		DMS	FMS	RMS
<b>System structure</b>		Fixed	Changeable	Changeable
<b>Machine structure</b>		Fixed	Fixed	Changeable
<b>Machine type</b>	<b>Production</b>	Manual machine	CNCs	RMTs and CNCs
	<b>Inspection</b>	Contact gauges	CMM	RIM
<b>System focus</b>		Single part	Machine	Part/product family
<b>Scalability</b>		No	Yes	Yes
<b>Flexibility</b>		No	Yes	Customized (around part family)
<b>Simultaneously operating tools</b>		Yes	No	Possible
<b>Productivity</b>		Very high	Low	High
<b>Cost per part</b>		<b>Low</b> for a single part when full capacity is used	<b>Reasonable</b> for several parts simultaneously, at low volume. Otherwise, <b>high</b>	<b>Medium</b> for producing part families at variable demand

Together, RMT and RIM allow the RMS not only to respond to new market demands in terms of product variety, but also in terms of product quality and accuracy, since RIM

allows an in-line inspection much faster than CMM machines, avoiding the production of faulty parts.

#### **1.4.1. Design of Reconfigurable Manufacturing Systems**

RMS must be designed with hardware and software modules that can be rapidly reconfigured and integrated to the system, in order to keep the reconfiguration process fast and practical (Koren et al. 1999). According to Koren (2010), the six key characteristics required to achieve that are: *Modularity*, *Convertibility*, *Scalability*, *Customization*, *Integrability*, and *Diagnosability*.

*Modularity* means that all main elements - software and hardware – are modular in order to meet the changeability requirements (Bi et al. 2008; Koren 2010). At the system level, each machine corresponds to a module, while at machine level, their components are modular (Koren et al. 1999; Koren 2010). *Convertibility* corresponds to the system ability to rapidly change its functionality to produce different products from the same product family (Bi et al. 2008; Koren 2010). At system level, it corresponds to integrate new machines, while at machine level it can be related to the change of machine modules or tools (Koren 2010).

*Scalability* is the system's ability to change its maximum production volume (Bi et al. 2008; Koren 2010). *Customization* means that machines from an RMS are manufactured around product/part families, rather than single parts (DMS) or any part (FMS) (Koren 2010). Therefore RMS has the required flexibility to manufacture these parts, allowing a substantial reduction on parts' costs while keeping a high productivity (Koren et al. 1999; Koren 2010). *Integrability* means that the components from the system are designed to be quickly integrated (Bi et al. 2008). Finally, *Diagnosability* is related to the system ability to identify system failures or faulty parts (Bi et al. 2008; Koren 2010).

According to Bi et al. (2008), there are three main critical issues related to the design of RMS: *architecture design*, *configuration design* and *control design*. *Architecture design*: is an activity related to the system design and is responsible to determine the system components and their interactions. *Configuration design*: is related to the system application and operation, being responsible for defining the system configuration of a specific system architecture to

perform a given activity, which can include planning or scheduling the production of a certain product family. In contrast, *Control design* is an activity related to the system operation. It is responsible for establishing the appropriate process variables, allowing the configured system to adequately perform the given activity.

In summary, Reconfigurable Manufacturing Systems' design process is much more complex than a fixed system due to its reconfiguration ability. Therefore, in addition to architecture design, it is also important to address process configuration and control when addressing the RMS design issues.

#### **1.4.2. Configuration selection in Reconfigurable Manufacturing Systems**

The configuration selection depends on the activities the system needs to perform. This is why the RMS configuration selection is often associated with the process or production planning. *Process* planning or process plan generation is the activity in which the product design data is transcribed into a method to fabricate a *single* product, including the specification of machines, tools as well as operation sequence (Nallakumarasamy et al. 2011; Mohapatra et al. 2013). *Production* planning is related to the selection of manufacturing resources to simultaneously fulfil multiple products' operations under some system constraints (e.g., resource capacity or availability) (Li and McMahon 2007).

The *process* plan represents the link between product and manufacturing systems since it transcribes the product features (or modules/components) into operations, which will be performed by various manufacturing resources (Najid et al. 2020). When the focus changes from one to multiple products and production and manufacturing system constraints (e.g., machine availability or capability, etc.) are considered, the bridge between products and manufacturing systems becomes the *production* plan.

In fixed systems, such as DMS, the process planning is accomplished once. However, the process planning must be constantly revisited in RMS because products and machines are constantly changing (ElMaraghy 2007). In addition to machines' configurations, companies also have to frequently redesign their RMS layout to adapt their system to manufacture new product variants in order to retain in the current unpredictable market (Maganha et al. 2019).

Layout reconfiguration/redesign is associated with deciding the machines' placements into the layout, often referred to as the "*machine layout problem*".

The new technologies provided by the 4<sup>th</sup> industrial revolution have increasingly allowed companies to redesign their RMS layouts within a short time and relatively low cost (Guan et al. 2012). However, machines' placement can still significantly impact manufacturing costs, lead times, and system productivity (Drira et al. 2007). Therefore, it is crucial to include the machine layout problem when optimizing the process or production planning in an RMS.

The configurability of machines individually joint to the layout configuration (machine layout problem) can considerably increase the complexity associated with the task of selecting the optimal RMS configuration – at system and machine levels simultaneously – for production or process planning. The RMS configuration design can be described as a combinatorial optimization problem, in which the decision-maker needs to decide which system configuration is the most appropriate for the current process requirements.

Considering that, a review about RMS configuration optimization was carried out. The following research bases: Scopus, ISI Web of Science, Taylor & Francis and Science Direct were consulted using the keywords "Reconfigurable Manufacturing System", "Configuration" and only journal papers addressing optimization problems were considered. This review was last updated on October 2020.

The fifty papers selected from the literature are presented in Table 5. Section 1.4.2.1 presents an overview of these papers, including MC issues and the integration of RMS's configuration with product decisions. Section 1.4.2.2 addresses the different RMS configuration-levels, which are summarized in Table 4. Section 1.4.2.3 analyzes the methods used to model and solve RMS configuration optimization problems.

Table 4. Description of the three main RMS configuration levels and their sub-categories in terms of layout design.

		With no layout design (N)	With layout design (LD)
System and machine level (SM)	<i>Machine configuration selection</i>	Machines and their configurations are selected with no reference to layout configuration issues	<ul style="list-style-type: none"> <li>- Machines and their configurations are selected together with the layout design;</li> <li>- Machines' configurations and their placement in stages or cells are addressed;</li> <li>- Machines and system reconfiguration associated to part/product family changes.</li> </ul>
System-level (SL)	<i>Machine selection only</i>	<ul style="list-style-type: none"> <li>- Multiple machine selection without considering layout design.</li> <li>- System configuration associated to a product/part family with no reference to machine configuration.</li> </ul>	<ul style="list-style-type: none"> <li>- Machine selection and its layout design, in which selected machines will be placed in a specific layout location (machine layout problem);</li> <li>- Machines placement into stages or cells is addressed, with no reference to machine's configuration;</li> <li>- Changeover in system configuration is addressed abstractly, with no reference to the machine's placement into the layout.</li> </ul>
Machine level (ML)	<i>Configuration selection for a single machine</i>	Configuration of a single machine; therefore, there is no consideration about layout design.	

Table 5 Summary of papers found in literature addressing RMS configuration optimization.

ID	Authors	Configuration-level	Machine layout problem	Process planning	Production planning	Product/Product family	Individual customer requirement	MC	Cost minimization										Other objectives optimized	
									System config. change	Capital cost	Raw material	Production operations	Material handling	Machine config. change	Tool using/changing	Layout design	Energy consumption	Maintenance		Inventory holding
1	Dou et al. (2020b)	SMLD			x															Tardiness
2	Dou et al. (2020a)	SMLD			x															Tardiness
3	Ghanei and Algeddawy (2020)	SLLD	x		x	x							x							-
4	Khezri et al. (2020)	SMN		x		x		x			x			x						Completion time; sustainability metric (wastes and GHG emissions)
5	Massimi et al. (2020)	SMN		x		x		x												Energy consumption
6	Singh et al. (2020)	SMLD			x				x	x								x		Reconfiguration time; lead-time; system availability, utilization, reliability and capability; system throughput; reconfiguration effort.
7	Touzout and Benyoucef (2019a)	SMN		x				x			x									Energy consumption, completion time
8	Touzout and Benyoucef (2019b)	SMN		x				x			x									Machine exploitation time, completion time
9	Benderbal and Benyoucef (2019)	SLLD	x	x																Machine utilization, presence of replacement machines in production line
10	Bortolini et al. (2019)	SMN			x						x									-
11	Liu et al. (2019)	ML			x					x										Cycle time
12	Moghaddam et al. (2019)	SMN			x					x										-
13	Asghar et al. (2018)	SMN		x																Machine capability
14	Ashraf and Hasan (2018)	SMLD			x															Operational capability, system re-configurability
15	Benderbal et al. (2018)	SMN		x				x												Completion time, system modularity
16	Moghaddam et al. (2018)	SMN			x					x										-
17	Pattanaik and Jena (2018)	SLN			x	x		x												Balance delay; cycle time; smoothness index
18	Battaia et al. (2016)	ML		x																





#### 1.4.2.1. Overview of papers found in literature

Table 5 shows that works focused on optimizing different objectives, such as completion time (Bensmaine et al. 2013; Benderbal et al. 2018; Touzout and Benyoucef 2019a), RMS key characteristics (Gupta et al. 2015; Goyal and Jain 2016; Koren et al. 2016; Singh et al. 2020), profit (Abbasi and Houshmand 2009; Abbasi and Houshmand 2011), etc. Not surprisingly, most of the papers have tried to minimize various kinds of costs. The capital cost and costs of operations and changing machines' configurations stood out as the most addressed. Nevertheless, few papers have addressed costs of layout design/configuration and material handling that can considerably affect the overall manufacturing cost.

Literature has shown an increasing interest in environmental aspects, since recent publications have addressed the minimization of energy consumption or costs associated with energy consumption, mainly based on the green house gas (GHG) emissions (Liu et al. 2019; Touzout and Benyoucef 2019a; Ghanei and Algeddawy 2020; Massimi et al. 2020). In addition to them, Khezri et al. (2020) have tried to optimize a sustainability metric that included wastes and GHG emissions.

Since RMS is a relatively new subject, there are still few papers considering MC issues when optimizing the RMS configuration. Some papers have cited MC as one of the main drivers of current manufacturing decisions, forcing companies to be more responsive, cost-time efficient, and sustainable (Touzout and Benyoucef 2019a; Touzout and Benyoucef 2019b; Khezri et al. 2020; Massimi et al. 2020).

Some papers focused on MC tried to select the best machines' set and process plan to maximize RMS's modularity or flexibility while minimizing the completion time to manufacture a single part/product from a family (Benderbal et al. 2017; Benderbal et al. 2018). Benderbal and Benyoucef (2019) focused on reducing the evolution effort within the same product family while keeping good performance in terms of the physical system layout to introduce new products for meeting the evolutions in customer requirements. Abdi and Labib (2016) investigated the impact of the product life cycle of different product families on the RMS's capacity usage. They addressed the impact of the customer needs changes on the product life cycle, based on market demands forecasting.

Other papers addressing MC tried to optimize the RMS configuration to attain optimal production planning capable of responding to a given end-product demand quantity (Dou et al. 2009a; Dou et al. 2010; Dou et al. 2011; Dou et al. 2016; Dou et al. 2020a). Pattanaik and Jena (2018), on the other hand, addressed the selection of customized modular products to be

assembled into a reconfigurable assembly system (RAS), a specific type of RMS limited to the assembly line. They reported the selection of modules and their assembly sequence to minimize balance delay, cycle time, and smoothness index.

Xu and Liang (2005, 2006) were not focused on MC, but they also tried to integrate modular products and assembly line configuration to optimize some performance criteria (e.g., product weight, reconfiguration cost, reliability, etc.). Yigit et al. (2002) and Yigit and Allahverdi (2003) concurrently optimized the modular product configuration with a RAS configuration, assuming that each module instance selected corresponded to a single RMT configuration. Although they did not address MC, they are focused on meeting individual/specific customer requirements. They stated many candidate sets of parameters, representing different customer requirements (e.g., price limits, desired quality levels, etc.), to constrain product configuration optimization depending on customer demands.

The variety of approaches previously presented for dealing with MC highlights the interest of research in RMS on addressing issues related to mass customizing products or dealing with high product variety. Most of the time, these works addressed RMS to manufacture a given product/part family. Indeed, RMS is known as a manufacturing system built around product/part families, explaining why most of the papers from Table 5 optimized the RMS configuration according to a product/part family.

There are still relatively few works addressing the concurrent optimization of product and RMS configurations, especially regarding MC. Papers that have done it mainly do not focus on meeting specific customer requirements.

#### **1.4.2.2. The RMS configuration-levels**

The RMS configuration can be classified into three main types: *machine-level (ML)*, *system-level (SL)* or *system and machine-levels (SM)*. The two last types can be subdivided into two categories *with layout* and *without layout* consideration. The description of each configuration level and its categories are detailed in Table 4.

Table 5 shows which papers addressed each of these different configuration-levels and categories that are machine-level (ML), system-level without layout consideration (SLN), system-level with layout design (SLLD), system and machine levels (SMN) without layout consideration as well as SM including layout design (SMLD) issues.

Battaia et al. (2016) and Liu et al. (2019) appear to be the only ones optimizing a single machine (ML) configuration to manufacturing different product families. Most of the papers from

SMN addressed machines' reconfigurations in terms of machines' modules and tools changes to generate the process plan of a single part/product from a family (Musharavati and Hamouda 2012; Bensmaine et al. 2013; Mohapatra et al. 2013; Benderbal et al. 2017; Asghar et al. 2018; Benderbal et al. 2018; Touzout and Benyoucef 2019b; Touzout and Benyoucef 2019a; Khezri et al. 2020; Massimi et al. 2020). Choi and Xirouchakis (2015), on the other hand, tried to optimize the production planning of an RMS to minimize energy consumption.

Other papers from SMN focused on minimizing the machines' reconfiguration costs to adjust the RMS's production capacity to respond to product demand fluctuations (Moghaddam et al. 2018; Moghaddam et al. 2019). In contrast, Bortolini et al. (2019) addressed a reconfigurable cellular manufacturing system (CMS), and they tried to reduce the total time associated with intercellular travels and machines' reconfigurations.

Eguia et al. (2017) also organized their RMS as a CMS; however, they addressed machine and layout configuration, being classified into the SMLD group. They split their problem into two steps: (1) *cell formation*, wherein they decided in which cell the CNCs and RMTs would be placed according to the process plan alternatives, and (2) *cell loading problem*, which consisted of determining the routing mix as well as the tool and module allocation. Although they addressed machine and layout configuration levels, they were not optimized concurrently.

Some works addressed the system's changeover costs of switching the manufacturing system from one part/product family to another. These costs included the addition/removal of machines or machines' modules (Gupta et al. 2014; Abdi and Labib 2016). The layout level is reconfigured due to the addition and removal of machines, but the machine layout problem is not addressed.

Other papers from SMLD addressed layout issues by considering a single or multi-part/product flow line in RMS, composed of various stages/workstations. These works mostly considered the operations' sequence, the number of parallel machines per stage, and machines' configurations as decisions to be made (Youssef and ElMaraghy 2006; Youssef and ElMaraghy 2007; Youssef and ElMaraghy 2008; Saxena and Jain 2012; Dou et al. 2020a; Dou et al. 2020b; Singh et al. 2020). Other papers selected machines' configurations and the number of parallel machines per workstations under a pre-defined operations sequence (Goyal et al. 2012; Goyal et al. 2013; Goyal and Jain 2016; Ashraf and Hasan 2018).

Some papers from SLLD also addressed the number of parallel machines per stage; however, they did not address the machines' configuration selection (Dou et al. 2009b; Li et al. 2009; Dou et al. 2010; Dou et al. 2011; Bryan et al. 2013; Dou et al. 2016). Wang and Koren (2012) and

Koren et al. (2016) focused on optimizing the system scalability by comparing different pre-fixed layout designs and the impact of adding machines to them to accommodate new product demands without considering the machine configuration level. Other works applied multi-criteria decision-making methods to select, among the set of pre-defined system configurations, the alternative providing the minimal capital cost and the maximum convertibility, scalability and productivity (Gupta et al. 2015), equipment availability, and reusability (Michalos et al. 2012).

Guan et al. (2012), Benderbal and Benyoucef (2019), and Ghanei and Algeddawy (2020) are the only papers from SLLD addressing the layout issues with a focus on the machine layout problem (Table 5). Benderbal and Benyoucef (2019) propose minimizing the evolution effort between products from the same family while keeping an optimal layout design, which is the one capable of satisfying all machine-location constraints. At first, they optimized the evolution effort by maximizing the presence of replacement machines in the production lines and minimizing the average use of machines as well as the layout evolution effort. The latter is quantified through the machines' similarities (base modules) and the different types of machines selected per product. In the second step, they minimized the penalties of non-satisfaction of location constraints, which refer to the minimum and maximum allowed distances between a pair of machines. They consider that RMTs can have different configurations by using this criterion to measure the machine similarity. Nevertheless, they did not specify the configuration assumed by each machine in their process plan, explaining why this work was classified as SLLD.

Guan et al. (2012) and Ghanei and Algeddawy (2020) modeled the layout design as the well-known Quadratic Assignment Problem (QAP). Since QAP is NP-hard (Sahni and Gonzalez 1976; Loiola et al. 2007), both works used stochastic methods such as revised electromagnetism-like mechanism (Guan et al. 2012) and genetic algorithm (GA) (Ghanei and Algeddawy 2020) to optimize layout configuration while minimizing material-handling costs. Some works that did not address layout issues but considered machine and system level (SMN) also addressed material-handling costs based on the time required to travel between two cells (Bortolini et al. 2019) or machines (Touzout and Benyoucef 2019b, a). These papers have shown the impact of material handling costs on the overall manufacturing costs.

Some papers from SLN addressed the system reconfiguration costs without addressing the machine layout problem. They assumed that each product/part family could have many associated system configurations (Abbasi and Houshmand 2009; Abbasi and Houshmand 2011). However, they did not specify whether RMS reconfiguration costs were related to changeovers in machines or layout configurations. Pattanaik and Jena (2018), on the other hand, tried to optimize the

selection of product modules and the sequence in which they should be assembled to minimize the overall balance delay, cycle time, and smoothness index.

Other works also addressed the selection of modular products together with the RMS configuration. However, they stated that the total system reconfiguration cost was associated with the module instances selected to be manufactured (Yigit et al. 2002; Yigit and Allahverdi 2003; Xu and Liang 2005; Xu and Liang 2006).

In summary, the RMS configuration optimization is mainly associated with the optimization of process or production planning, as shown in Table 5. Most of the papers addressed the RMS configuration-level at machine and layout. The machines' reconfiguration costs were one of the main costs addressed by works. In contrast, few papers addressed layout reconfiguration/redesign costs. Although many works considered the layout configuration level, few papers focused on the machine layout problem (i.e., deciding the machines' placement into the layout). Further, relatively few papers addressed material handling costs, which can be highly affected by the layout design (Maganha et al. 2019).

#### **1.4.2.3. Modelling and optimizing the RMS configuration**

Table 6 summarizes the approaches used to model and solve the optimization problems related to the RMS configuration. Each number correspond to the ID number of each publication presented in Table 5.

Papers addressing the RMS configuration optimization were mostly multi-variate and multi-objective, evidencing the complexity of choosing the most appropriate RMS configuration. The RMS configuration optimization problems were mainly modelled with integer variables varying between linear and nonlinear models, but problems modelled as multi-criteria decision-making problems were also found. The "undefined model" means papers that did not specify the mathematical modelling type of their problems.

Nonlinear problems were solved mainly by non-deterministic/stochastic approaches, with emphasis on GA-based algorithm that was sometimes applied singly (Abbasi & Houshmand, 2011; Dou et al., 2010, 2011; Wang & Koren, 2012) or hybridized with other methods, such as dynamic programming (Bryan et al. 2013) and tabu search (Youssef and ElMaraghy 2007; Youssef and ElMaraghy 2008).

Table 6 Approaches used to model and solve optimization problems of RMS configuration.

Techniques		Models		Linear			Nonlinear		MCDM	Undefined model
				LP	ILP	MILP	MINLP	NLIP		
Deterministic approaches	CKSP		42							
	epsilon-constraint based method			1, 2; 4						
	AHP						6			
	Topological sort								36	
	weighted sum	25								
	Others		7					6		
Non-deterministic approaches	Metaheuristics	NSGA-II		7; 8	1; 2; 4; 11				28; 31	
		GA			3	37; 38	39; 40; 46; 48		13; 24; 26; 47	
		MOPSO	23		2				30	
		SA/AMOSA/MOSA		7	11					9; 35
		TS					41			
		SPEA			4					
	Heuristics	Shannon entropy						6; 26		
		Intelligent search						34		
		Other heuristics		8	18	5; 33				17
	Hybrids	(NSGA-II or AMOSA) + TOPSIS								14; 15; 19; 32
		GA + other algorithms		43				29		44; 45
		Markov analysis + Decision tree								21
Software and Solvers (CPLEX / Gurobi / LINGO / GAMS)			10; 12	3; 12; 16; 20; 22		39; 46; 49				
Undefined method						50				

**Models:** ILP: Integer Linear Programming; MILP: Mixed integer linear programming; MINLP: Mixed Integer Non-Linear Programming; NLIP: Nonlinear integer programming; MCDM: Multi-Criteria Decision-Making problems.

**Techniques:** AHP: Analytic Hierarchy Process; CKSP: Constrained K-shortest path; GA: Genetic Algorithm; NSGA-II: Non-dominated sorting GA; SA: Simulated annealing; AMOSA: Archived Multi-Objective SA; MOPSO: Multiple Objective Particle Swarm Optimization; TOPSIS: Technique for Order of Preference by Similarity to Ideal Solution.

A commercial software such as LINGO, CPLEX, and Gurobi is based on deterministic approaches and has a good ability to solve mixed and integer linear problems. Due to that, some works tried to validate the results they obtained by non-deterministic algorithms by comparing them with those obtained by these commercial solvers (Battaia et al., 2016; Dou et al., 2011, 2016; Xu & Liang, 2006). These solvers were also used singly for solving integer and mixed-integer linear problems (Eguia et al. 2017; Bortolini et al. 2019; Moghaddam et al. 2019).

Non-deterministic methods were mainly applied for solving RMS configuration optimization problems than deterministic ones. There is no consensus about the best optimization method for solving RMS configuration problems because it depends on the problem addressed and how it is addressed (e.g., machine and/or layout configuration levels). However, there is an evident increasing interest in applying metaheuristic methods for optimizing RMS configuration problems, especially GA-based approaches.

Metaheuristics is the junction of *meta* “beyond” or “higher level” with *heuristic* that means “to find” or “to discover by trial and error”, being capable of obtaining good quality solutions – with a higher performance than heuristics - for a complex problem in a reasonable time with no guarantee of finding the optimal (Gandomi et al. 2013). Since they are not problem-specific, metaheuristics have the ability to solve numerous optimization problems with relatively few modifications in the algorithm (Blum and Roli 2003). Hence, complex optimization problems (i.e. NP-hard and NP-complete) can be solved by these methods, even if it is unknown how to translate the whole problem into a mathematical programming model. These advantages may explain why metaheuristics have been increasingly applied in RMS researches.

### **1.5. Integrating configuration decisions from modular product and process**

Integrated product and process design (IPPD) is a management strategy in which all essential activities from product design and manufacturing are integrated, thanks to a multidisciplinary team that works to optimize decisions to meet performance objectives (US Department of Defense 1996; Mervyn et al. 2004). IPPD seeks to reduce lead-time and cost while improving product quality and customer satisfaction (Mervyn et al. 2004).

As previously presented, the modular design and RMS design englobes different steps, such as module definition, architecture formation, and configuration. At the configuration phase, the variety of products occurs, while in RMS, the configuration allows the system to change/adapt its functionality to fulfill other processing operations to manufacture other parts. Therefore, simultaneously configuring both can increase the overall system’s (i.e., product plus process) flexibility and responsiveness.

Product modularity has a strong relationship with process and resources’ modularity (Kusiak 2002). Hence, to improve performance criteria such as cost, time, reliability, etc., product and process should be optimized simultaneously. Designing modular products without considering the process issues (e.g., production/assembly line constraints, current configuration, etc.) may lead to additional costs and time; while if the process is configured



without considering product requirements, it may not be able to accomplish the desired product functionalities (Xu and Liang 2005; Xu and Liang 2006).

In MC contexts, defining the product configuration and process decisions (e.g., production or process planning, etc.) concurrently is a key issue in the customer and manufacturer relationship, becoming a crucial factor of successful MC implementation (Pitiot et al. 2013; Pitiot et al. 2014). Nevertheless, both product configuration and process planning are often considered in a sequence manner, meaning the product is configured, and then its process planning is established (Pitiot et al. 2013).

Many researchers dealing with classical manufacturing or assembly systems (other than RMS) have tried to integrate process planning or individual customer requirements with the product configuration. Works focused on MC integrate customer requirements into the product configuration decision, but they often do not address process decisions.

Wang et al. (2017b), for example, suggest a small interval of cost for each product feature, while other works consider that each product module or component compounding the product has a fixed associated cost (Lee et al. 2001; Li et al. 2006; Yang and Dong 2012; Yang et al. 2020). Although they consider customer requirements, these works do not directly address process decisions (i.e., process/production planning, scheduling, etc.) because they assume that each module/component has a fixed cost with no reference to changes in the process.

Customer requirements are also often integrated into the modular product configuration (or product module selection) optimization in assemble-to-order environments since this strategy is based on assembling pre-fabricated components/modules after customer orders are received (Da Cunha et al. 2007; Khalaf et al. 2010b; Khalaf et al. 2011a; Khalaf et al. 2011b). These works are generally focused on selecting the supplier of components/subassemblies that will lead to minimal cost while respecting the customer's deadline and requirements when the product is assembled in a single assembly plant.

Customer requirements are also often addressed in the literature as functions/functional requirements that are mapped into modules or module instances (Chen and Lin 2002; Xie et al. 2005; Li et al. 2006; Khalaf et al. 2010a; Khalaf et al. 2011b; Wang et al. 2017a; Yang et al. 2020). These works consider that each customer requirement corresponds to a function associated with a given module, meaning that if this module appears in the final product, then the function (customer requirement) is satisfied. It means that the product configuration optimization was constrained by the functional requirements required by the customer.

Some works concurrently optimizing the product configuration with the process planning for MC addressed customer requirements, instead of functions, by dividing them into two categories: non-negotiable (must be respected) and negotiable requirements (Pitiot et al. 2013, 2014, 2020). They mapped their problem into a constraint satisfaction problem (CSP) by dividing it into two steps. In the first one, they filtered the solution space with non-negotiable elements, and in the second step, they optimized the process planning integrated to the product configuration according to the negotiable requirements.

Wang et al. (2017a) optimized the process configuration to manufacturing mass-customized product variants according to specific customer requirements. They mapped their problem as a generative CSP by addressing constraints at the process level (e.g., temporal constraints between activities, resources constraints, etc.) and the product level (e.g., material, thickness, and hole of some product feature, as well as tolerance requirements).

In summary, works optimizing product configuration based on specific customer requirements generally do it by associating a function to a corresponding customer requirement that can be directly accomplished by a module present in the final product. However, in some cases, customer requirements are addressed directly, especially for works using a CSP approach. The process decisions with which product configuration is generally integrated are process or assembly planning, in which the focus is on defining the operations sequence and their assignment to the available resources.

### **1.5.1. Integrating configuration decisions of modular product and RMS**

Section 1.5 presented some research working on the concurrent optimization of product and process decisions driven by specific customer requirements, highlighting the relevance of integrating decisions from product and process for manufacturing mass-customized products.

This sub-section focuses on researches that concurrently optimized the product configuration with process decisions in an RMS. A detailed literature review was carried out through two research databases: ISI Web of Science (WoS) from Thomsom Reuters and Scopus from Elsevier Science, which are currently the two main academic research databases, according to Arezoo et al. (2013). The keywords used for the product level were “modular product” OR “product configuration” OR “module selection” OR “selection of module”, which were combined with the following process keywords: “reconfigurable” AND (“manufacturing system” OR “assembly system” OR “assembly line”). The “advanced search” was used for collecting papers in both databases. While product and process keywords were searched on the WoS with the operator

“TS”, meaning topic search, on the Scopus search, the operator “TITLE-ABS-KEY” was used.

All searches were limited to documents written in *English language*, resulting in a total of 32 publications, from which 11 were found in WoS and 21 in Scopus. A manual screening was applied to remove the 10 duplicates found within the databases, resulting in 22 publications. Finally, each of the 22 publications’ abstracts was carefully read, and 12 papers were selected, as presented Table 7. This research was last updated on March 2021.

### **1.5.2. Overview of papers found in literature**

There are few works addressing both subjects in an integrated manner. This can be explained by the fact that RMS is a relatively new subject.

Many papers addressed MC. Besides them, Pattanaik and Jena (2018) presented an approach to assemble customized modular products. Other works proposed approaches to scheduling or planning modular products in RMS, highlighting the advantages to exploiting the modularity and flexibility of both levels (product and process) to manufacture mass-customized products (Ye and Liang 2005; Xu and Liang 2006; Ye and Liang 2006).

Some works addressing MC were focused on quality control under the high variety demand of products manufactured in reconfigurable systems. Hassan et al. (2011) used a simulation environment to implement a hybrid manufacturing cell to deal with constant product changeovers. They used a quality control system to inspect the quality of parts produced. Besides that, they also used a priority coefficient for scheduling the customer orders. Davrajh and Bright (2013), on the other hand, developed a reconfigurable apparatus to measure product configuration quality according to individual customer qualitative requirements for each product module compounding a product variant (e.g., color, diameter, size, etc.).

Besides the previous papers, others that did not address MC were also focused on individual customer requirements. Yigit et al. (2002) and Yigit and Allahverdi (2003), for example, minimized the average quality loss and the manufacturing reconfiguration costs for producing modular products according to the set of parameters chosen by the customers (e.g., capability, desired quality level, price limits etc.). These parameters were weighted according to the order priorities, customer importance and demands.

Bryan et al. (2007) addressed individual customer requirements for developing a co-evolution of product families and assembly systems. They used an approach to design and

reconfigure a product family evolving over time, based on customer requirements, concurrently with the assembly system configuration. Therefore, they are more focused on product family formation and reconfiguration over time than on a single product configuration at a time.

Since RMS is developed around a product/part family, many researchers have investigated the optimization of product family formation/design or evolution considering RMS reconfiguration issues, as detailed in Section 1.4 (Bryan et al. 2013; Gupta et al. 2014; Abdi and Labib 2016; Benderbal and Benyoucef 2019). However, differing from Bryan et al. (2007), these works did not address the module selection for reconfiguring or creating product families; they addressed only the product variants of families instead.

Table 7. Papers selected from the literature review.

<b>Authors</b>	<b>Individual Customer requirements</b>	<b>Mass Customization</b>	<b>Optimization</b>
Pattanaik and Jena (2018)		x	x
Mittal et al. (2018)			x
Müller et al. (2016)			
Davrajh and Bright (2013)	x	x	
Hassan et al. (2011)	x	x	
Ye and Liang (2006)		x	x
Xu and Liang (2006)		x	x
Xu and Liang (2005)			x
Ye and Liang (2005)		x	x
Yigit and Allahverdi (2003)	x		x
Yigit et al. (2002)	x		x
Bryan et al. (2007)	x		x

Müller et al. (2016) developed an object-oriented reference model to describe the product and process requirements for supporting the assembly planning engineer and information provision. They connected product and process information by describing the requirements for assembly from the view of the product.

Most of the papers addressing the configuration of modular products and RMS focused on optimizing them concurrently. As previously explained, the configuration of modular products or reconfigurable processes can be described as a combinatorial optimization problem. It means, given a set of available elements (e.g., product modules, machines, machines' modules, configurations, etc.) the decision-maker must choose which ones will be selected to fulfill the needs. The next subsection will detail how the literature has addressed

the concurrent optimization of modular products and RMS configurations, including the RMS configuration-level issues.

### **1.5.3. Concurrent optimization problems: modelling and solving**

Table 8 suggests that most of the papers focused on dealing with high product variety since they all considered that the same module could have many variants/options (module instances). The presence of module instances/variants increases product variants' quantity due to the higher number of module combinations. Thus, considering module instances can increase the optimization problem complexity, especially when product configuration decisions are integrated with the RMS ones.

Limiting the product configuration decision to module selection, with no consideration of module instances, could be a strategy to reduce the optimization problem complexity. Nevertheless, in the context of highly mass-customized products, it is essential to consider multiple variants because this is the only way to have enough product variety to meet each customer's uniqueness.

Most of the papers included the module instance selection as a decision variable of their product configuration optimization problems to know whether an instance was selected for a given module (Xu and Liang 2005; Xu and Liang 2006) for a particular product variant (Bryan et al. 2007) or to accomplish a parameter set (Yigit et al. 2002; Yigit and Allahverdi 2003). Mittal et al. (2018) used multi-criteria decision analysis to choose a product variant according to the production process in an RMS, based on the instances/options of modules present in each product variant.

Pattanaik and Jena (2018), on the other hand, do not consider module instance selection as a decision variable in their optimization problem. Instead, they considered multiple product variants, each composed of different combinations of module instances and having different alternative assembly times depending on their module instances and the sequence the latter were assembled. The alternative assembly times were used to minimize the balance delay, smoothness index, and cycle time.

Ye and Liang (2005, 2006) appear to be the only ones integrating modular product scheduling with cell configuration decisions in a cellular RMS. They addressed the scheduling of different product variants simultaneously by assigning the jobs required by their module instances to different manufacturing cells. They considered that each cell was responsible for producing all options/instances of a given module.

The literature presented in Section 1.4 has shown that papers focused on optimizing the RMS configuration singly usually address system and machine level, especially recent publications. When decisions concerning RMS configuration are integrated with product configuration decisions, the papers usually address only the system configuration level, with no reference to machine reconfiguration. Mittal et al. (2018) appear to be the only work addressing machine reconfiguration issues by considering the effort of adding/removing modules from reconfigurable machines. They did not address layout design issues.

Ye and Liang (2005, 2006) addressed layout design issues in a cellular RMS. They stated that the system configuration changeovers were based on machines' position changes within each manufacturing cell, with no reference to reconfigurations at machine level. They stated a fixed set of feasible cell configurations to do each job from which a given configuration is selected. Therefore, once the cell configuration is selected, it is possible to know the machines' positions, and then the material handling costs can be calculated.

In addition to them, Bryan et al. (2007) also addressed layout reconfiguration. Instead of cells, they considered a single-line assembly, with multiple stations that could be changed over different product family generations depending on the increasing product demands. If future generations of product families need higher system capacities, parallel stations could be added to each current layout station. They did not consider the reconfiguration issues at the machine level.

All other papers addressed system level without including layout configuration issues (Yigit et al. 2002; Yigit and Allahverdi 2003; Xu and Liang 2005; Xu and Liang 2006). These works considered that the system reconfiguration costs were associated with the selected product module instances without specifying whether these costs are associated with changing machine or layout configuration.

Different process decisions were considered when the works integrated the optimization of product and RMS configurations. Most of the works tried to optimize the product configuration with its assembly line balancing (Xu and Liang 2005; Xu and Liang 2006; Bryan et al. 2007; Pattanaik and Jena 2018) or process planning (Yigit et al. 2002; Yigit and Allahverdi 2003).

Most of the papers tried to minimize costs of reconfiguration, operations (Yigit et al. 2002; Yigit and Allahverdi 2003; Xu and Liang 2005; Xu and Liang 2006; Mittal et al. 2018), material handling and work-in-process (Ye and Liang 2005; Ye and Liang 2006). Some papers also focused on minimizing the smoothness index (Xu and Liang 2005; Pattanaik and Jena 2018), as well as cycle time, balance delay (Pattanaik and Jena 2018), and idle time (Xu and Liang 2006). Bryan et al. (2007), in turn, maximized the profit. Some papers also focused on optimizing

characteristics associated with the product, such as power and reliability (Xu and Liang 2005; Mittal et al. 2018) and the average quality loss (Yigit et al. 2002; Yigit and Allahverdi 2003; Xu and Liang 2006).

Product configuration, process planning, and machine layout design are typically combinatorial optimization problems, meaning that there are a finite set of available (integer) elements, and the focus is on finding an optimal object (or combination of elements) among them (Papadimitrou and Steiglitz 1998). Not surprisingly, almost all papers described their optimization problems using mathematical nonlinear integer programming (NLIP) models, except for Mittal et al. (2018), who made a multi-criteria decision analysis using the analytic hierarchy process (AHP) methodology.

GA stood out as the most used stochastic method for solving nonlinear problems, being applied for solving both uni and multi-objective optimization problems (Xu and Liang 2005; Ye and Liang 2005; Xu and Liang 2006; Ye and Liang 2006; Bryan et al. 2007). Some of them tried to validate the results obtained by GA by comparing them with outputs provided by LINGO, a commercial software capable of solving nonlinear integer problems (Xu and Liang 2005; Ye and Liang 2005; Xu and Liang 2006; Ye and Liang 2006).

This literature review shows that some works have already focused on concurrently optimizing the modular product and RMS configurations. However, there is still no work addressing the concurrent optimization of product configuration (through the module instance selection) with the process planning (including machine reconfigurability) and machine layout design in RMS, driven by individual customer requirements.

Table 8. Papers optimizing the configuration of modular products and RMS concurrently.

Authors	System Configuration -level	Module instances	Process decision	Objective functions		Optimization methods to :	
				Min	Max	Model	Solve
Pattanaik and Jena (2018)	SLN	x	ALP	Balance delay; cycle time; smoothness index		NLIP	Heuristic method
Mittal et al. (2018)	SMN	x	PP	Reconfiguration cost	Product reliability and power	MCDM	AHP
Ye and Liang (2006)	SLLD	x	Scheduling	Cost(cell and system reconfiguration; operations; material handling; work in process; machine idle)		NLIP	GA/LINGO
Xu and Liang (2006)	SLN	x	ALB	Cost (raw material, system reconfiguration and operations); Quality loss function; idle time		NLIP	GA/LINGO
Xu and Liang (2005)	SLN	x	ALB	Cost (raw material, system reconfiguration and operations); Smoothness index	Reliability	NLIP	GA
Ye and Liang (2005)	SLLD	x	Scheduling	Cost(cell and system reconfiguration; operations; material handling; work in process; machine idle)		NLIP	GA/LINGO
Yigit and Allahverdi (2003)	SLN	x	PP	Average quality loss; reconfiguration cost		NLIP	LINGO
Yigit et al. (2002)	SLN	x	PP	Average quality loss; reconfiguration cost		NLIP	-
Bryan et al. (2007)	SLLD	x	ALB		Profit	NLIP	GA

Assembly line balancing (ALB); Nonlinear integer programming (NLIP); Process planning (PP); System-level with no layout design (SLN); System level with layout design (SLLD); System and Machine without layout design issues (SMN);.



## **1.6. Final considerations from this chapter**

In recent years, MC has evolved to adapt and respond to new customer demands. MC is currently on its advanced stage, in which the focus is no more on satisfying market segments' needs but on meeting each individual customer requirements. The production of highly customized or personalized products in the current manufacturing era, characterized by unpredictable and diverse market demand, requires a responsive business model focused on the customer while keeping a closer interaction between the product development and the manufacturing system.

Modular design is already known as the best strategy to offer high product variety. Producing modules relatively independent that different products can share allows increasing the product variety while reducing lead time, contributing to increasing the economy of scale. In MC contexts, customers can be integrated into the value creation through the product configuration phase. Customers can decide, among the proposed options, which modules and module instances they will choose to configure a product that will meet their uniqueness.

However, when the customer integration is not done correctly, the vast number of options can lead to the “mass confusion” paradigm, driving customers' dissatisfaction since they become unable to choose what they want. This reinforces the importance of integrating customers during product development and helping them to find the most appropriate product configuration capable of fitting their unique needs. Besides the customer integration into the product configuration, another key issue in the customer and manufacturer relationship in MC is the concurrent decision of product configuration and process planning.

Modular products and processes have a strong interaction since a product designed with no consideration of process constraints may cause high costs and time, directly impacting customer satisfaction. A process configured without consideration of the product configuration may not accomplish the functionalities required by the product. Therefore, optimizing product and process in a sequential manner, instead of concurrently optimizing both, can considerably offset the potential of using modular design in product and process. That highlights the relevance of integrating product and process configuration for MC.

Besides all modular manufacturing systems, the RMS is the most appropriate for coping with the current changeable market due to its high responsiveness. RMS can quickly adjust its capacity and functionality to produce new products from the same family for responding to new market demands. Hence, manufacturing modular products in RMS seems to be the

best strategy to cope with the challenges of the advanced level of MC or mass personalization. Modular products and RMS together can increase the whole system flexibility and responsiveness, enabling the advantage stage of MC, as illustrated in Figure 8.

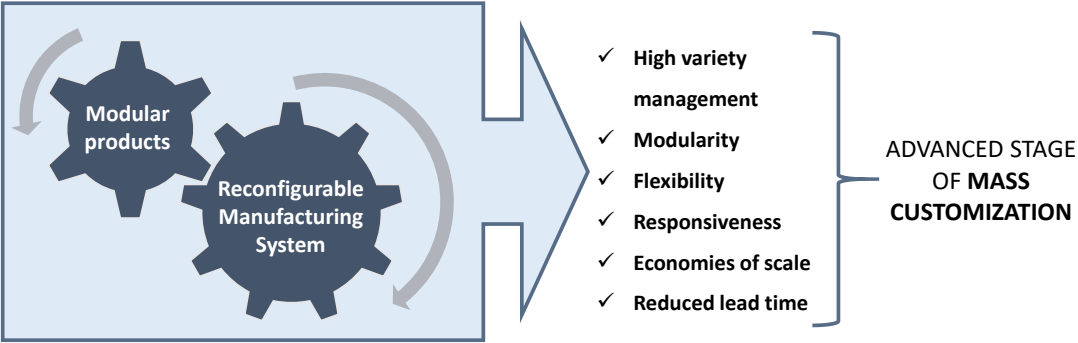


Figure 8. Impact of coupling modular product design strategy to the RMS on mass customization.

While modular product design has already been widely investigated in literature and is well known as the most appropriate product design strategy for MC, RMS is relatively a new manufacturing system that has been increasingly addressed in the literature. The literature review presented in this chapter shows that, although many papers are optimizing product configuration or RMS configuration separately, there are still very few papers addressing both issues concurrently.

Some works determine a fixed RMS configuration cost for each module instance selected to compound the product variant. However, they did not define if whether the reconfiguration costs arise from the layout or machines reconfiguration. Also, papers addressing only RMS configuration (presented in Section 1.4.2) most of the time have not yet addressed layout configuration issues, such as machine displacement and material handling, which can affect the overall manufacturing costs.

The current changeable and unpredictable market demands have forced companies to constantly change the RMS configuration at machine and layout levels to respond to new product requirements. Hence, it also becomes necessary to further investigate the RMS's machine layout problem when configuring products for MC.

The literature has shown that there are still few works from RMS addressing MC issues. Hence, the number of works optimizing product configuration and RMS configuration for MC while considering individual customer requirements is still fewer.

The main gaps found in this literature are summarized as following:

- Product configuration:
  - Researches addressing product configuration for MC are mainly focused on meeting individual customer requirements in terms of desired product functionalities or characteristics. However, most of the time, they did not consider process issues/constraints;
  - Many researchers have embedded the production costs into each module instance's cost, assuming that it will always be the same, no matter the production process (e.g., operations sequence and assignment).
- RMS configuration:
  - Few papers optimizing the RMS configuration addressed the MC;
  - Customer demands are only addressed by papers optimizing the production planning to fabricate products according to a deterministic demand quantity and a delivery date established by the customers;
  - Most of the papers optimizing the RMS configuration address the machine and system levels. Several papers addressed configuration issues at the machine level, but few of them considered the machine layout problem;
  - Few papers addressed the machine configuration issues and the machine layout problem. The works addressing both issues did not do it concurrently;
  - Few papers have directly addressed the initial/current configuration of machines and layout before configuring an RMS to generate a process or production plan;
- Integrated product and RMS configurations:
  - There are still very few works optimizing the product configuration concurrently with the process planning in RMS. Hence, there are even

fewer works concurrently optimizing the product configuration to meet individual customer requirements in MC contexts;

- Most of the works addressing concurrent optimization of product and RMS configurations address the RMS at the system configuration level, without including neither machine nor layout reconfiguration issues.

Considering the review presented in this chapter and the gaps highlighted above, it is possible to conclude that there is still work to be done on the concurrent optimization of modular product configuration and process planning as well as machine layout configuration in RMS for MC. In the following chapter, two propositions will be presented in order to fill these research gaps.

## CHAPTER 2 - CONCEPTUAL AND MATHEMATICAL MODELS

It has been shown that concurrently optimizing modular product and RMS configurations is a key step to produce highly mass-customized products for meeting customer uniqueness effectively. Therefore, it seems to be relevant developing new methods to concurrently optimize the product configuration, driven by individual customer requirements, with the RMS configuration.

In order to contribute to the literature of the integrated configuration of modular products and reconfigurable systems for mass customization (MC), this chapter brings two propositions to model and solve the optimization of modular product and RMS process configurations concurrently.

Proposition 1 consists of the concurrent optimization of product and process planning in an RMS. Proposition 2 is an evolution of proposition 1 since, in addition to the product configuration and process planning, the machine layout design is also addressed. Both propositions are detailed as follows.

***Proposition 1: Concurrent optimization of modular product configuration and process planning in RMS:***

*What:*

- Concurrent optimization of the product configuration with the process planning in an RMS, driven by individual customer requirements. The current machines' configurations will be taken into account as well as the distance between them;

*How:*

- An Integer Linear Programming (ILP) model is proposed to minimize the overall manufacturing costs, including costs of i) raw material, ii) operations iii) machines' reconfiguration, and iv) material handling.

*Why:*

- Optimizing concurrently the product configuration and the process planning in an RMS can lead to lower total costs;

- Few works have addressed the optimization of product and process planning in RMS. However, decisions from both are closely related and should be considered concurrently;
- Even fewer works optimizing both decisions concurrently focused on meeting individual customer requirements for MC;
- There are few works considering the initial machines' configuration in RMS when optimizing the process planning, even when only RMS configuration is addressed (i.e., product configuration issues are not considered);
- There are relatively few works addressing costs associated with material handling in RMS;
- There is only one work addressing the concurrent optimization of modular product configuration and the process planning in an RMS, by considering the machine reconfigurability (Mittal et al. 2018).

***Proposition 2 (evolution of proposition 1): (Concurrent optimization of modular product configuration and process planning and layout configuration in RMS:***

*What:*

- Concurrent optimization of the product configuration with the process planning and machine layout configuration in an RMS, driven by individual customer requirements. The current machines' and layout configurations will be considered.

*How:*

- A Nonlinear Integer Linear Programming (NLIP) model is proposed to minimize the overall manufacturing costs, including costs of i) raw material, ii) operations iii) machines' reconfiguration, and iv) material handling, and iv) layout reconfiguration (machine displacement).

*Why:*

Besides the other reasons previously presented:

- There are very few works addressing the optimization of product and process planning and the machine layout configuration in RMS;

- Currently, there is no work integrating the optimization of modular product and RMS configurations by considering that the RMS can be switched within the production of two different products from the same family, in the same generation;
- The layout design is still a subject little explored in RMS. Few papers have addressed the reconfiguration of layout in RMS even when only RMS configuration is addressed (i.e., product configuration issues are not considered);
- There is still no work focused on optimizing concurrently the modular product configuration and the RMS configuration, at machine and system levels, while considering the machine layout problem.

In summary, this work presents two propositions to concurrently optimize the product configuration, driven by individual customer requirements, with the 1) process planning in an RMS and with the 2) process planning and machine layout design in RMS.

Before detailing propositions 1 and 2 in Section 2.2, Section 2.1 will explain the main assumptions made in this work to clarify the proposal.

## **2.1. Explaining the main assumptions**

### **2.1.1. Individual customer requirements**

In products with a modular architecture, customer requirements (CRs) are translated into functional requirements/functions (FRs), which is then mapped into design parameters (DPs), which are fulfilled by product modules/components. Therefore, in the product configuration phase, the relation between CRs, FRs, and DPs is already known. As presented in Chapter 1, many works configuring products according to specific customer requirements used the strategy of assuming that each CR is directly correspondent to an FR, which is fulfilled by a given module or module instance in the product variant. Most of the time, papers consider a one-to-one mapping from FRs to modules, meaning that each FR is directly fulfilled by a single module instance present in each product variant.

Since CRs have correspondent FRs, and the latter are mapped into modules, the FRs can constrain the optimization of the product configuration problem, ensuring that all individual customer requirements will be satisfied by the configured product. Considering the advantages of addressing CRs as FRs for constraining the product configuration optimization, this work adopted this approach.

### **2.1.2. Operations identification**

In this work, the product configuration is integrated with the RMS configuration. Therefore, the operations to be manufactured are not previously known because they will be defined depending on the configuration of the product selected to be manufactured. Each module selected to compound a product is supposed to require a set of operations, which is a parameter in these optimization problems. Hence, operations selection/identification means just identifying operations to be fulfilled according to the module instances selected to compound the product.

### **2.1.3. Machine configuration**

In this work, the machine configuration level is considered, meaning that machines can assume multiple configurations, but once at a time. As presented in Chapter 1, the machines' reconfiguration in RMS can be represented by generic configurations per machine (i.e., C1, C2, etc.) or by means of changes in a base or alternative modules as well as tools. Changing modules and tools also requires the machine's software reconfiguration. In this work, the machine configuration is addressed in a generic way, meaning that a given configuration can include a specific module and tool as well as a specific setup in the software. Therefore, changes in machine configuration can mean changes in hardware and software. The reconfiguration cost and time will be proportional to the kind of reconfiguration. It means, if only software reconfiguration is required, the setup process will be faster and cheaper than when hardware changeovers are also needed.

### **2.1.4. Process planning**

Process planning or process plan generation means the operations sequence as well as the assignment of operations to specific machine-configurations. This work does not address the performance criteria associated with each machine, such as reliability and capability. Therefore, machines' performance criteria are assumed to be all 100%.

### **2.1.5. Layout configuration**

In this work, it is assumed that there is a  $n$  number of available layout locations equal to the  $n$  number of available machines. The distance between each pair of layout locations is known. Each layout location is assumed to be big enough to place machines without overlapping while respecting security distances. Therefore, the layout configuration here means deciding in which available layout locations each machine will be placed considering



the current layout configuration (machine layout problem). Figure 9 illustrates the layout reconfiguration through an example of a multi-rows layout with fixed locations.

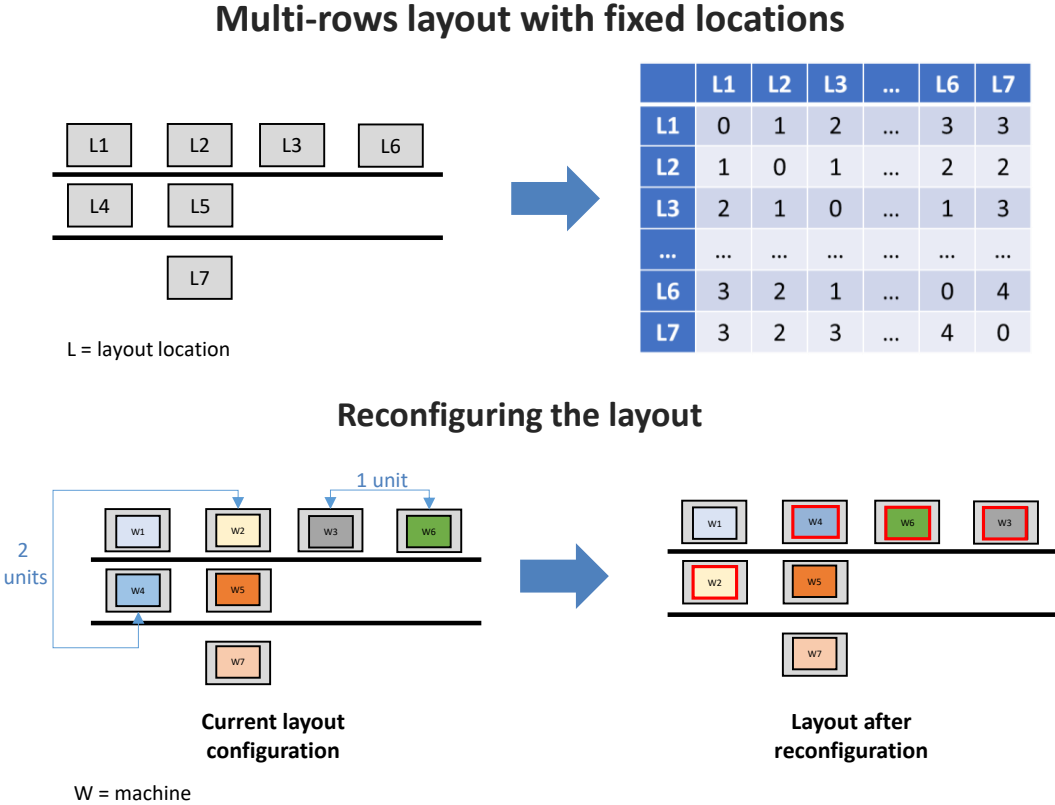


Figure 9. Illustration of layout reconfiguration.

**2.2. Mathematical model propositions**

As previously explained, the approach adopted in this work assumes that CRs can be directly associated with FRs, and each FR can be satisfied by one or multiple available module instances. FRs here represent the functional characteristics desired by the customer in the final product, while each module instance is a basic “building block” from which the product is assembled.

The customization process starts with the customer choosing from all available FRs, which ones she/he desires in her/his product. Each FR can be satisfied by at least one module instance available, meaning that many product variants can emerge from the combination of different module instances available to satisfy the same set of FRs required by the customer.

Figure 10 illustrates the concurrent optimization of modular product and RMS configurations, driven by individual customer requirements. Once the customer selected the desired FRs in the product, the latter will constraint the solution space. The solution space of

product configuration comprises all feasible combinations of module instances (product variants) according to their compatibility.

Module instances belonging to each product variant require specific operations to be fabricated and assembled, which are supposed to be sequenced and assigned to different machine-configurations while ensuring minimal processing cost. Therefore, each product variant candidate requires a given RMS configuration. Figure 10 shows that in proposition 1 the solution space of RMS is limited to the machine configuration selection because it addresses only the process planning. In contrast, in proposition 2, besides the machine selection, the RMS solution space also includes the layout configuration selection.

In both propositions, it is assumed that each machine can assume multiple configurations and perform different operations, while different machine-configurations can perform the same operation. The optimal solution will correspond to the product configuration and i) process planning (proposition 1) or ii) process planning and layout configuration (proposition 2) that together led to the minimal overall manufacturing cost, including raw material, operations, machine reconfiguration, material handling and layout reconfiguration (only for proposition 2).

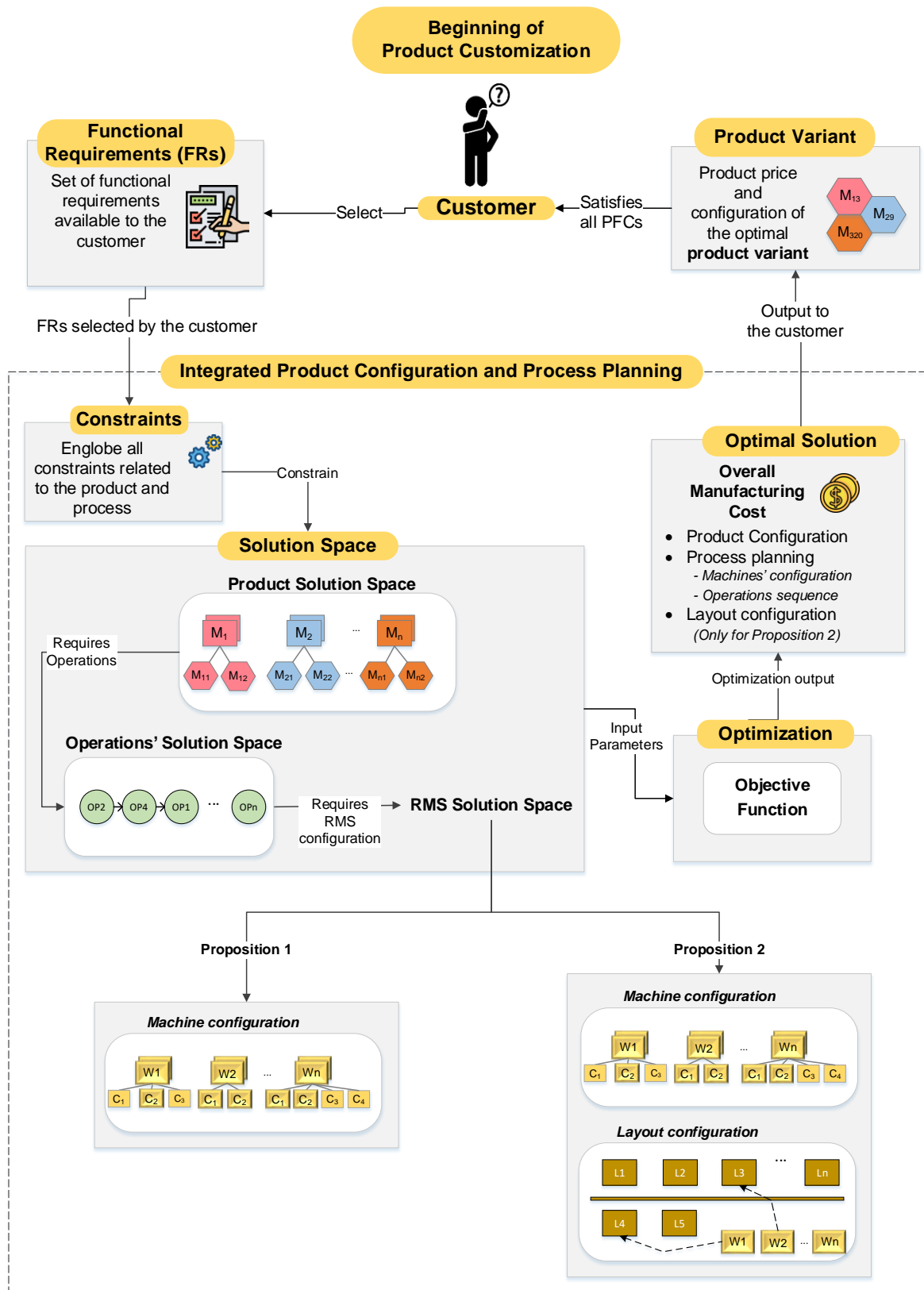


Figure 10. Conceptual model of the concurrent optimization of modular product and RMS configurations, driven by individual customer requirements.

## 2.2.1. Proposition 1

### 2.2.1.1. ILP model

This section introduces the ILP model to concurrently optimize the modular product configuration and process planning in an RMS, driven by individual customer requirements. The objective of this ILP model is the minimization of the overall manufacturing costs, which include costs of (1) module instances' raw material (CRM), (2) manufacturing and assembly operations (COP), (3) machine configuration change (CCC) and (4) material-handling (MHC).

*Assumptions:*

- I. For each available function, there is at least one available module instance capable of satisfying it;
- II. Module instances are independent, meaning that the selection of an instance from module  $m$  does not imply the inclusion of another instance from module  $m'$ .
- III. Modules have decoupled interfaces, meaning that changes in one module do not affect another one;
- IV. The raw material for producing module instances are prompt available;
- V. Cost and time of changing an RMT configuration include both hardware and software levels;
- VI. All machines are reconfigurable. Each machine can be configured in a set of available configurations associated with each machine type;
- VII. Each machine configuration can perform one or more operations;
- VIII. All machines are available and are compatible;
- IX. No operational performance criteria (e.g., machine capability, reliability, etc.) is addressed; thus, all of them are supposed to be 100%;
- X. Each operation can be performed by one or several pairs of machine-configuration.

Based on the assumptions presented above, the major decisions related to the integrated optimization of the product's and RMS's configuration are described as follows:

- I. Decisions on product configuration:
  - a. *Module instances selection:* define which module instances will compose the product variant, ensuring that all FRs will be satisfied.

- b. Operations identification: identify which operations will be manufactured based on the module instances selected to compose the product variant.

II. Decisions on RMS configuration selection:

- a. Process planning: sequence and assign the identified operations to the machine-configurations while respecting precedence and compatibility constraints as well as considering the current machine configuration.

*Notations:*

$f$	Index of all available FRs
$m, m'$	Module index
$i, i'$	Module instance index
$p, p'$	Operations index
$w, w'$	Machine index
$c, c'$	Machine configuration index
$j$	Index of process plan position

*Input parameters:*

Some information must be available to allow the problem resolution:

- I. Product and FRs: the set of modules/module instances available to fulfill the available functions in the product are provided by the product design team. No module instance can be created in this step of product configuration.

---

$F$	$= \{1, 2, \dots, f, \dots,  F \}$ , Set of all available functions (or FRs)
$D_f$	$\begin{cases} = 1, & \text{if } f \text{ is required} \\ = 0, & \text{otherwise} \end{cases}$

---

$M$	$= \{1, 2, \dots, m, \dots,  M \}$ , Set of available modules
$I$	$= \{1, 2, \dots, i, \dots,  I \}$ , Set of available module instances
$G_{m,i,m',i'}$	$\begin{cases} = 1, & \text{if instance } i \text{ of module } m \text{ is compatible with instance } i' \text{ of module } m' \\ = 0, & \text{otherwise} \end{cases}$
$S_{mif}$	$\begin{cases} = 1, & \text{if instance } i \text{ of module } m \text{ satisfies the function } f \\ = 0, & \text{otherwise} \end{cases}$

---

- II. Product processing: the information regarding the operations required by each product module instance and the precedence relation between these operations are known.

---

$OP$	$= \{1, 2, \dots, p, \dots,  OP \}$ , Set of available operations
$R_{pmi}$	$\begin{cases} = 1, \text{if instance } i \text{ of module } m \text{ requires an operation } p \text{ to be} \\ \text{manufactured} \\ = 0, \text{otherwise} \end{cases}$
$P_{pp'}$	$\begin{cases} = 1, \text{if an operation } p \text{ must be processed before an operation } p' \\ \forall p, p' \in OP, \quad p \neq p' \\ = 0, \text{otherwise} \end{cases}$

---

III. Process planning: the machine-configurations cost and time required to accomplish each operation  $p$  is supposed to be available. The cost and time of changing machines' configuration and handling material are also available. Machines are placed into layout locations that have known distances between them.

---

$W$	$= \{1, 2, \dots, w, \dots,  W \}$ , Set of all available machines
$C_w$	$= \{1, 2, \dots, c, \dots,  C_w \}$ , Set of available configurations for machine $w$
$c_w^0$	Initial configuration $c^0$ of machine $w$ , where $c_w^0 \in C_w$
$Q_{pwc}$	$\begin{cases} = 1, \text{if an operation } p \text{ is feasible on machine } w \text{ with} \\ \text{configuration } c \\ = 0, \text{otherwise} \end{cases}$

---

$J$	$= \{1, 2, \dots, j, \dots,  J \}$ , Set of all process plan positions
-----	--

---

$d_{ww'}$	Distance between each pair ( $w$ and $w'$ ) of machines
-----------	---

---

$CR_{mi}$	Raw material cost of an instance $i$ of module $m$
$CO_{pwc}$	Cost of operation $p$ processed on machine $w$ with configuration $c$ per time unit
$CC_{w,cc'}$	Cost of machine $w$ changing from configuration $c$ to $c'$ per time unit
$CT$	Cost of transporting material per distance unit

---

$TO_{pwc}$	Time of processing operation $p$ on machine $w$ with configuration $c$
$TC_{w,cc'}$	Time of machine $w$ changing from configuration $c$ to $c'$

---

Decision variables:

All decisions variables of the ILP model are binary, as described as follows:

$$y_{mi} \quad \begin{cases} = 1, \text{if instance } i \text{ of module } m \text{ is selected} \\ = 0, \text{otherwise} \end{cases}$$

$$x_{pwcj} \quad \begin{cases} = 1, \text{if operation } p \text{ is assigned to machine } w \text{ with} \\ \text{configuration } c \text{ in process plan position } j \\ = 0, \text{otherwise} \end{cases}$$

$$\begin{aligned}
u_{w,c,c',j,j+1} & \begin{cases} = 1, \text{ if there is a change of configuration } c \text{ to } c' \text{ of machine} \\ \qquad \qquad \qquad w, \text{ between position } j \text{ and } j + 1 \\ \qquad \qquad \qquad \text{ in the process plan} \\ = 0, \text{ otherwise} \end{cases} \\
h_{wcj} & \begin{cases} = 1, \text{ if machine } w \text{ with configuration } c \text{ is in process plan} \\ \qquad \qquad \qquad \text{ position } j \\ = 0, \text{ otherwise} \end{cases} \\
t_{w,w',j,j+1} & \begin{cases} = 1, \text{ if machine } w \text{ is utilized at position } j \text{ and machine} \\ \qquad \qquad \qquad w' \text{ is utilized at position } j + 1 \text{ in the process plan} \\ = 0, \text{ otherwise} \end{cases}
\end{aligned}$$

*Objective function:*

As previously stated, this mathematical model's objective is to minimize the overall manufacturing cost, including costs of raw material, operations, machine configuration change and material handling. Eq. (1) presents the objective function.

$$\begin{aligned}
z &= f^{CRM} + f^{COP} + f^{CCC} + f^{CMH} \\
\min z &
\end{aligned} \tag{1}$$

*Where:*

$f^{CRM}$ : **Cost of raw material** of module instances:

$$\sum_m \sum_i y_{mi} \times CR_{mi} \tag{2}$$

$f^{COP}$ : **Cost of operations** (manufacturing and assembly) carried out each in a given machine-configuration pair:

$$\sum_j \sum_p \sum_w \sum_c x_{pwcj} \times CO_{pwc} \times TO_{pwc} \tag{3}$$

$f^{CCC}$ : **Cost of changing the configuration** of machines:

$$\sum_w \sum_{c,c' \in C_w} \sum_j u_{w,cc',jj+1} \times CC_{w,cc'} \times TC_{w,cc'} \tag{4}$$

$f^{CMH}$ : **Costs of material handling** between all pairs of machines:

$$\sum_{w,w' \in W} \sum_j t_{ww'j,j+1} \times CT \times d_{w,w'} \quad (5)$$

*Subject to:*

The following constraints are related to the product configuration decision. Eq.(6) states that only one instance  $i$  of module  $m$  can be selected from alternative instances of the same module at a time. Eq.(7) ensures that only compatible module instances will be present in the same product variant. Eq.(8) ensures that each required PF must be satisfied by at least one selected module instance at a time.

$$\sum_i y_{mi} \leq 1 \quad \forall m \in M \quad (6)$$

$$y_{mi} + y_{m'i'} \leq G_{m,i,m',i'} + 1 \quad \begin{array}{l} \forall m, m' \in M \\ \forall i, i' \in I \end{array} \quad (7)$$

$$\sum_m \sum_i y_{mi} \times S_{mif} \geq D_f \quad \forall f \in F \quad (8)$$

The operations related constraints are stated by the following equations. Eq.(9) states that each operation is processed at most at one process plan position by one machine-configuration, while Eq. (10) states that each operation is processed at most once in one machine-configuration. With constraints Eq.(9) and (10), Eq. (11) ensures that only required operations will be processed, and they must each be performed in only one machine-configuration in only one process plan position. Eq. (12) states that an operation can only run in a machine having the correct configuration. Operations' precedence relationships are stated by Eq. (13).

$$\sum_p \sum_w \sum_c x_{pwcj} \leq 1 \quad \forall j \in J \quad (9)$$

$$\sum_w \sum_c \sum_j x_{pwcj} \leq 1 \quad \forall p \in OP \quad (10)$$



$$\sum_p \sum_m \sum_i y_{mi} \times R_{pmi} = \sum_p \sum_w \sum_c \sum_j x_{pwcj} \times Q_{pwc} \quad (11)$$

$$x_{pwcj} \leq h_{wcj} \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \\ \forall c \in C \end{array} \quad (12)$$

$$\sum_{w'} \sum_{j'=1}^{j-1} x_{p'w'c'j'} \geq \sum_w \sum_c x_{p,w,c,j} \times P_{p'rp} \quad \begin{array}{l} \forall p, p' \in OP \\ p \neq p' \\ \forall j = 1, 2, \dots, n \end{array} \quad (13)$$

The following equations concern the constraints mainly associated to the machine changes. Eq. (14) ensures that there is at most one configuration change for a machine  $w$  between position  $j - 1$  and  $j$ . Eq. (15) ensures that there is at most one machine change between position  $j - 1$  and  $j$ . Eq. (16) states if there is a machine change between position  $j - 1$  and  $j$ , while Eq. (17) states if there is a configuration change for a given machine  $w$  between position  $j - 1$  and  $j$ . Eq.(18) states that each machine has exactly one configuration at each process plan position. Eq. (19) ensures that a machine's configuration stays the same unless its configuration is changed in the process plan. Eq. (20) sets the initial configuration of each machine.

$$\sum_{c,c' \in C_w} u_{w,c,c',j,j+1} \leq 1 \quad \begin{array}{l} \forall j \in J \\ \forall w \in W \end{array} \quad (14)$$

$$\sum_{w,w' \in W} t_{w,w',j,j+1} \leq 1 \quad \forall j \in J \quad (15)$$

$$t_{w,w',j,j+1} \geq \sum_c (x_{pwcj} + x_{p'w'c',j+1}) - 1 \quad \begin{array}{l} \forall j \in J \\ \forall p, p' \in OP \\ \forall w, w' \in W \\ w \neq w' \end{array} \quad (16)$$

$$u_{w,c,c',j,j+1} \geq x_{pwcj} + x_{p'w'c',j+1} - 1 \quad \begin{array}{l} \forall j \in J \\ \forall p, p' \in OP \\ \forall w \in W \\ \forall c, c' \in C_w \\ c \neq c' \end{array} \quad (17)$$

$$\sum_c h_{wcj} = 1 \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \end{array} \quad (18)$$

$$h_{wcj} = h_{wcj-1} + \sum_{c'} u_{w,c',c,j-1,j} - \sum_{c'} u_{w,c,c'j-1,j} \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \\ \forall c \in C \end{array} \quad (19)$$

$$h_{wcj} = c_w^0 \quad \begin{array}{l} \forall w \in W \\ \forall c \in C \\ \forall j = 0 \end{array} \quad (20)$$

Finally, Eq.(21)-(25) represent the decision variables domains.

$$y_{mi} \in \{0,1\} \quad \forall m \in M \quad \forall i \in I \quad (21)$$

$$x_{pwcj} \in \{0,1\} \quad \begin{array}{l} \forall w \in W \\ \forall p \in OP \end{array} \quad \begin{array}{l} \forall j \in J \\ \forall c \in C_w \end{array} \quad (22)$$

$$u_{j,j+1,w,c,c'} \in \{0,1\} \quad \begin{array}{l} \forall j \in J \\ \forall c, c' \in C_w \end{array} \quad \forall w \in W \quad (23)$$

$$t_{w,w',j,j+1} \in \{0,1\} \quad \forall j \in J \quad \forall w, w' \in W \quad (24)$$

$$h_{wcj} \in \{0,1\} \quad \begin{array}{l} \forall w \in W \\ \forall c \in C \end{array} \quad \forall j \in J \quad (25)$$

### 2.2.2. Proposition 2

In addition to integrating decisions of modular product configuration with the process planning, this work also investigates the influence of layout reconfiguration on the overall manufacturing costs. This is why proposition 2 includes the layout (re)configuration issues into the decision, representing an evolution of proposition 1.

As previously stated, the operations to be performed depend on the module instances selected to compound the feasible/candidate product variants. The operations sequence and assignment will depend on the machine-configurations selected to perform them and the layout configuration to reduce costs of reconfiguring machines, operating, or handling the material. It is important to highlight that the choice of machines' configurations and machine locations into the layout will depend on the current configuration of machines and layout.

The material handling costs are strongly dependent on the machines' positions in the layout. Depending on the operations required by a given product variant, another layout configuration can be better than the current one in terms of material handling costs minimization. However, reconfiguring layout (i.e., displacing machines into the layout) implies additional manufacturing costs. Therefore, one crucial decision is to identify whether

there is a good trade-off between changing the layout configuration and minimizing the overall manufacturing cost.

Since this proposition is an evolution of proposition 1, some assumptions, and equations of objective function and constraints are quite similar. Hence, the NLIP mathematical model's main differences (i.e., assumptions, notations, parameters, decision variables and equations) regarding the ILP one will be highlighted in bold.

#### 2.2.2.1. NLIP model

This section introduces the NLIP model to concurrently optimize the product configuration with the process planning and layout configuration in an RMS, driven by individual customer requirements. The objective of this NLIP model is to minimize the overall manufacturing cost, including the following costs (1) CRM, (2) COP, (3) CCC, (4) CMH, and **(5) layout reconfiguration (CLR)**. In order to reduce the complexity associated with this problem, the following assumptions were stated:

*Assumptions:*

- I. For each available function, there is at least one available module instance capable of satisfying it;
- II. Module instances are independent, meaning that the selection of an instance from module  $m$  does not imply the inclusion of another instance from module  $m'$ ;
- III. Modules present decoupled interfaces, meaning that changes in one module do not affect another one;
- IV. The raw-material for producing module instances are prompt available;
- V. Cost and time of changing an RMT configuration include both hardware and software levels;
- VI. All machines are reconfigurable. Each machine can be configured in a set of available configurations associated with each machine type;
- VII. Each machine configuration can perform one or more operations;
- VIII. All machines are available and are compatible;
- IX. No operational performance criteria (e.g., machine capability, reliability, etc.) is addressed; thus, all of them are supposed to be 100%;
- X. Each operation can be performed by one or several pairs of machine-configuration.

- XI. The layout is composed by fixed locations and the distances between them are known;**
- XII. All layout locations have the same size and format and they are big enough to place a machine without overlapping, while respecting security distances;**
- XIII. All machines can be placed in all layout locations;**
- XIV. The displacing machines' cost and time comprise the machine installation and uninstallation and its displacement on the floor.**

Considering the assumptions presented above, the principal decisions associated to the NLIP problem are described as follows:

1. Decisions on product configuration:
  - a. Module instances selection: define which module instances will compose the product variant, ensuring that all FRs will be satisfied.
  - b. Operations identification: identify which operations will be manufactured based on the module instances selected to compose the product variant.

The decisions presented above are exactly the same presented for Proposition 1. Therefore, there is no difference between both models/propositions regarding decisions at the product configuration.

2. Decisions on RMS configuration selection:
  - a. Process planning: sequence and assign the identified operations to the machine-configurations while respecting precedence and compatibility constraints as well as considering the current machine configuration.
  - b. Layout configuration: select a layout location to place each machine while considering the current layout configuration.

Since material handling is dependent on the travelled distance between machines, besides the costs of displacing machines, layout (re)configuration can considerably affect material handling costs.

*Notations:*

$f$  Index of all available FRs

$m, m'$	Module index
$i, i'$	Module instance index
$p, p'$	Operations index
$w, w'$	Machine index
$c, c'$	Machine configuration index
$j$	Index of process plan position
$k, k'$	<b>Index of layout location</b>

### *Input parameters*

There is some information that must be available for allowing the problem resolution.

- I. Product and functions: the set of modules/module instances available to fulfil the available functions in the product are provided by the product design team. No module instance can be created in this step of product configuration.

$F = \{1, 2, \dots, f, \dots, |F|\}$  Set of all available functions (or FRs)

$D_f = \begin{cases} = 1, & \text{if } f \text{ is required} \\ = 0, & \text{otherwise} \end{cases}$

$M = \{1, 2, \dots, m, \dots, |M|\}$  Set of available modules

$I = \{1, 2, \dots, i, \dots, |I|\}$  Set of available module instances

$G_{m,i,m',i'} = \begin{cases} = 1, & \text{if instance } i \text{ of module } m \text{ is compatible with instance} \\ & \text{ } i' \text{ of module } m' \\ = 0, & \text{otherwise} \end{cases}$

$S_{mif} = \begin{cases} = 1, & \text{if instance } i \text{ of module } m \text{ satisfies the function } f \\ = 0, & \text{otherwise} \end{cases}$

- II. Product processing: the information regarding the operations required by each product module instance and the precedence relation between these operations are known.

$OP = \{1, 2, \dots, p, \dots, |OP|\}$ , Set of available operations

$R_{pmi} = \begin{cases} = 1, & \text{if instance } i \text{ of module } m \text{ requires an operation } p \text{ to be} \\ & \text{ } \text{manufactured} \\ = 0, & \text{otherwise} \end{cases}$

$P_{pp'} = \begin{cases} = 1, & \text{if an operation } p \text{ must be processed before an operation } p' \\ & \text{ } \forall p, p' \in OP, p \neq p' \\ = 0, & \text{otherwise} \end{cases}$

- III. RMS configuration information for process planning and layout design: the machine-configurations cost and time required to accomplish each operation  $p$  are supposed

to be available. Further, the time and cost of changing machines' configuration, displacing machines, and transporting material are also available.

$W$	$= \{1, 2, \dots, w, \dots,  W \}$	Set of all available machines
$C_w$	$= \{1, 2, \dots, c, \dots,  C_w \}$	Set of available configurations for machine $w$
$c_w^0$		Initial configuration $c^0$ of machine $w$ , where $c_w^0 \in C_w$
$k_w^0$		<b>Initial position <math>k^0</math> of machine <math>w</math> into the layout</b>
$Q_{pwc}$	$\begin{cases} = 1, \text{ if an operation } p \text{ is feasible on machine } w \text{ with} \\ \text{configuration } c \\ = 0, \text{ otherwise} \end{cases}$	
$J$	$= \{1, 2, \dots, j, \dots,  J \}$	Set of all process plan positions
$d_{kk'}$		<b>Distance between each pair (<math>k</math> and <math>k'</math>) of layout locations</b>
$CR_{mi}$		Raw material cost of an instance $i$ of module $m$
$CO_{pwc}$		Cost of operation $p$ processed on machine $w$ with configuration $c$ per time unit
$CC_{w,cc'}$		Cost of machine $w$ changing from configuration $c$ to $c'$ per time unit
$CT$		Cost of transporting material per distance unit
$CD_w$		<b>Cost of displacing machine <math>w</math> per time unit</b>
$TO_{pwc}$		Time of processing operation $p$ on machine $w$ with configuration $c$
$TC_{w,cc'}$		Time of machine $w$ changing from configuration $c$ to $c'$
$TD_w$		<b>Time per distance unit for displacing machine <math>w</math></b>

*Decision variables:*

All decision variables of the NLIP model are binary as described below.

$$y_{mi} \begin{cases} = 1, \text{ if instance } i \text{ of module } m \text{ is selected} \\ = 0, \text{ otherwise} \end{cases}$$

$$x_{pwcj} \begin{cases} = 1, \text{ if operation } p \text{ is assigned to machine } w \text{ with} \\ \text{configuration } c \text{ in process plan position } j \\ = 0, \text{ otherwise} \end{cases}$$

$$u_{w,c,c',j,j+1} \begin{cases} = 1, \text{ if there is a change of configuration } c \text{ to } c' \text{ of machine} \\ w, \text{ between position } j \text{ and } j + 1 \\ \text{in the process plan} \\ = 0, \text{ otherwise} \end{cases}$$

$$h_{wcj} \begin{cases} = 1, \text{ if machine } w \text{ with configuration } c \text{ is in process plan} \\ \text{position } j \\ = 0, \text{ otherwise} \end{cases}$$

$$t_{w,w',j,j+1} \begin{cases} = 1, \text{ if machine } w \text{ is utilized at position } j \text{ and machine} \\ w' \text{ is utilized at position } j + 1 \text{ in the process plan} \\ = 0, \text{ otherwise} \end{cases}$$

$$\ell_{wk} \begin{cases} = 1, \text{ if machine } w \text{ is placed on the layout location } k \\ = 0, \text{ otherwise} \end{cases}$$

*Objective function:*

As previously stated, the objective of this NLIP model is to minimize the overall manufacturing cost related to product configuration, process planning, and layout configuration in an RMS, which is described by the objective function presented in Eq.(26).

$$z = f^{CMI} + f^{COP} + f^{CCC} + f^{CMH} + f^{CLR} \quad (26)$$

*min z*

*Where:*

$f^{CRM}$ : **Cost of raw material** of module instances:

$$\sum_m \sum_i y_{mi} \times CR_{mi} \quad (27)$$

$f^{COP}$ : **Cost of operations** (manufacturing and assembly) carried out each in a given machine-configuration pair:

$$\sum_j \sum_p \sum_w \sum_c x_{pwcj} \times CO_{pwc} \times TO_{pwc} \quad (28)$$

$f^{CCC}$ : **Cost of changing the configuration** of machines:

$$\sum_w \sum_{c,c' \in C_w} \sum_j u_{w,cc',j-1,j} \times CC_{w,cc'} \times TC_{w,cc'} \quad (29)$$

$f^{CMH}$ : **Cost of material handling** between all pairs of machines:

$$\sum_{w,w' \in W} \sum_{k,k' \in K} \sum_j t_{ww'j-1,j} \times \ell_{wk} \times \ell_{w'k'} \times CT \times d_{k,k'} \quad (30)$$

$f^{CLR}$ : **Cost of layout reconfiguration** from the current layout configuration to the optimal one:

$$\sum_w \sum_k \ell_{wk} \times CD_w \times TD_w \times d_{kk'} \quad (31)$$

Where here  $k' = k_w^0$ , meaning that distances between two layout locations are measured only in relation to the initial position  $k^0$  and the final position  $k$  of machine  $w$ .

*Subject to:*

The following constraints are related to the product configuration problem. Eq.(32) states that only one instance  $i$  of module  $m$  can be selected from alternative instances of the same module at a time. Eq.(33), in turn, ensures that only compatible module instances will be present in the same product variant. Finally, Eq.(34) ensures that each required PF must be satisfied by one selected module instance at a time.

$$\sum_i y_{mi} \leq 1 \quad \forall m \in M \quad (32)$$

$$y_{mi} + y_{m'i'} \leq G_{m,i,m',i'} + 1 \quad \forall m, m' \in M \quad (33)$$

$$\sum_m \sum_i y_{mi} \times S_{mif} \geq D_f \quad \forall f \in F \quad (34)$$

The operations related constraints are represented by the following equations. Eq.(35) states that each operation is processed at most at one process plan position by one machine-configuration, while Eq.(36) states that each operation is processed at most once in one machine-configuration. With constraints (35) and (36), Eq.(38) ensures that only operations required by the selected module instances will be performed, and each must be accomplished by only one machine-configuration in only one process plan stage. Eq. (37) states that an operation can only run in a machine that has the correct configuration and constraint Eq.(39) states the operations' precedence relationships.

$$\sum_p \sum_w \sum_c x_{pwcj} \leq 1 \quad \forall j \in J \quad (35)$$

$$\sum_w \sum_c \sum_j x_{pwcj} \leq 1 \quad \forall p \in OP \quad (36)$$



$$x_{pwcj} \leq h_{wcj} \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \end{array} \quad \begin{array}{l} \forall c \in C \\ \forall p \in OP \end{array} \quad (37)$$

$$\sum_p \sum_m \sum_i y_{mi} \times R_{pmi} = \sum_p \sum_w \sum_c \sum_j x_{pwcj} \times Q_{pwc} \quad (38)$$

$$\sum_{w'} \sum_{j'=1}^{j-1} x_{p'w'c'j'} \geq \sum_w \sum_c x_{p,w,c,j} \times P_{p'p} \quad \begin{array}{l} \forall p, p' \in OP \\ p \neq p' \\ \forall j \in J \end{array} \quad (39)$$

The following constraints are associated to the machine configuration level. Eq.(40) ensures that there is at most one configuration change for a machine  $w$  between position  $j - 1$  and  $j$ , whilst Eq.(41) ensures that there is at most one machine change between position  $j - 1$  and  $j$ . Eq.(42) states if there is a machine change between position  $j - 1$  and  $j$ , while Eq.(43) states if there is a configuration change for a given machine  $w$  between positions  $j - 1$  and  $j$ . Eq. (44) states that each machine has exactly one configuration at each process plan position. Eq. (45) ensures that the configuration of a machine stays the same unless its configuration is changed in the process plan. Eq.(46) sets the initial configuration of each machine.

$$\sum_{c,c' \in C_w} u_{w,c,c',j-1,j} \leq 1 \quad \begin{array}{l} \forall j \in J \\ \forall w \in W \end{array} \quad (40)$$

$$\sum_{w,w' \in W} t_{w,w',j-1,j} \leq 1 \quad \forall j \in J \quad (41)$$

$$t_{w,w',j-1,j} \geq \sum_c (x_{pwcj-1} + x_{p'w'cj}) - 1 \quad \begin{array}{l} \forall j \in J \\ \forall p, p' \in OP \\ \forall w, w' \in W \\ w \neq w' \end{array} \quad (42)$$

$$u_{w,c,c',j-1,j} \geq x_{pwcj-1} + x_{p'w'c'j} - 1 \quad \begin{array}{l} \forall j \in J \\ \forall p, p' \in OP \\ \forall w \in W \end{array} \quad \begin{array}{l} \forall c, c' \in C_w \\ c \neq c' \end{array} \quad (43)$$

$$\sum_c h_{wcj} = 1 \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \end{array} \quad (44)$$

$$h_{wcj} = h_{wcj-1} + \sum_{c'} u_{w,c',c,j-1,j} - \sum_{c'} u_{w,c,c',j-1,j} \quad \begin{array}{l} \forall w \in W \\ \forall j \in J \\ \forall c \in C \end{array} \quad (45)$$

$$h_{wcj} = c_w^0 \quad \begin{array}{l} \forall w \in W \\ \forall c \in C \\ \forall j = 0 \end{array} \quad (46)$$

The following equations represent constraints of the layout configuration level. Eq. (47) ensures that each layout location ( $k$ ) should have one machine ( $w$ ) at a time, while Eq. (48) ensures that each machine ( $w$ ) should be placed in only one layout location ( $k$ ) at a time. With constraints (47) and (48), Eq. (49) ensures that the number of layout locations occupied will be equal to the number of available machines.

$$\sum_w \ell_{wk} = 1 \quad \forall k \in K \quad (47)$$

$$\sum_k \ell_{wk} = 1 \quad \forall w \in W \quad (48)$$

$$\sum_w \sum_k \ell_{wk} = |W| \quad (49)$$

The decisions variables' domains are stated by Eq.(50)-(55).

$$y_{mi} \in \{0,1\} \quad \forall m \in M \quad \forall i \in I \quad (50)$$

$$x_{pwcj} \in \{0,1\} \quad \begin{array}{l} \forall w \in W \\ \forall p \in OP \end{array} \quad \begin{array}{l} \forall j \in J \\ \forall c \in C_w \end{array} \quad (51)$$

$$u_{j,j+1,w,c,c'} \in \{0,1\} \quad \begin{array}{l} \forall j \in J \\ \forall c, c' \in C_w \end{array} \quad \forall w \in W \quad (52)$$

$$t_{w,w',j,j+1} \in \{0,1\} \quad \forall j \in J \quad \forall w, w' \in W \quad (53)$$

$$h_{wcj} \in \{0,1\} \quad \begin{array}{l} \forall w \in W \\ \forall c \in C \end{array} \quad \forall j \in J \quad (54)$$

$$\ell_{wk} \in \{0,1\} \quad \forall w \in W \quad \forall k \in K \quad (55)$$

### 2.3. Final considerations from this chapter

Based on the gaps highlighted in the literature review, this chapter presented two propositions to concurrently optimize modular products and RMS configuration decisions while considering specific customer requirements. Both approaches aim to minimize the overall manufacturing cost, including costs of i) raw material, ii) operations, iii) machine configuration change, and iv) material handling. Proposition 2 is an evolution of proposition 1, and besides the previous costs, it also considers the layout reconfiguration cost.

The approaches proposed here differ from those found in the literature addressing the concurrent optimization of product and RMS configurations because here, the RMS

reconfiguration costs are not fixed according to the type of module instance selected. Instead, product and process decisions are integrated through the operations required by the selected module instances. The machine-configurations selected to manufacture a given product variant depend on the different operations, which vary according to the types of module instances compounding it. The selected machine-configurations also depend on the current machines' configurations (propositions 1 and 2) and layout configuration (proposition 2).

It is assumed that different machines' configurations can perform the same operation, and the same machine-configuration can perform many operations. Therefore, a given product variant's operations cost can vary depending on the machine and configurations chose to manufacture it. Due to the possibility of assigning operations to different machine-configurations while respecting precedence constraints, there are different possibilities of sequencing the process, which implies a variable material handling cost as well.

Proposition 2 also considers the layout configuration, meaning that in addition to decide in which machine-configuration each operation will be fulfilled and in which sequence, it is crucial to decide if the machines should change their current layout position. Since there are still few works addressing the machine layout problem in RMS, and even fewer simultaneously considering the machine configuration level and the layout, proposition 2 also aims to contribute to the RMS layout design field.

In summary, both propositions try to deal with the scenarios of highly mass-customized products, focusing on meeting specific customer requirements. Their objective is to find the optimal product variant, in terms of cost, to be manufactured in the current RMS configuration (including machine and layout levels). The choice of the optimal product variant depends, simultaneously, on the raw material costs and its *process* plan. The latter can be used posteriorly as a base for optimizing the production planning, in which process capacity constraints and many customer orders will be considered simultaneously.

Although both propositions present some advantages to calculate the product configuration costs according to the RMS configurability, they have some limitations. One limitation is related to the operations. Since required operations per module instance englobe manufacturing and assembly ones, the latter does not depend on the pairs of module instances to be assembled.

Besides that, the machine configuration level is generically addressed since it englobes software and hardware configurations, with no reference to the machine's base-module and changeable-modules or tools. In proposition 2, the layout is supposed to be completely reconfigurable, meaning that all machines can be easily displaced among the pre-defined layout locations without considering layout obstacles. Proposition 2 does not address immobile machines, but it can handle them by considering high displacing costs. However, if all machines cannot displace in an RMS, proposition 1 seems to be more appropriate because it is a linear model, requiring less computation effort compared to the NLIP model (proposition 2).

At the product level, it is assumed that each functional requirement (FR) is entirely satisfied by a single or many module instance in the product, and module instances are partially independent. This means that selecting a given module instance does not imply the selection of another one, meaning that the inclusion relationships are not considered. The only product configuration rules considered in this work are the module instances compatibility and instances of the same module restriction. Other configuration rules can be considered in future research such as inclusion.

Customer requirements are addressed in terms of FRs. They constrain the optimization problem to ensure that the product variant selected satisfies all FRs with the minimal cost. However, customer satisfaction may vary depending on the module instances compounding the product. Between two product variants capable of satisfying the customer's needs, there is one that she/he may prefer, being ready to pay for that even if it is more expensive than the other option. Therefore, in future research, the customer satisfaction index can be measured and optimized concurrently with the overall manufacturing cost.

## CHAPTER 3 - SOLUTION APPROACHES

### 3.1. Solution approaches proposition

GA-based approaches have proven their ability to solve configuration selection problems for modular products (Li et al. 2006; Yang et al. 2020) or RMS (Youssef and ElMaraghy 2007; Youssef and ElMaraghy 2008; Goyal et al. 2012). The literature review presented in Chapter 2 showed that genetic algorithm (GA) stood out as the most used method to solve the concurrent optimization of product and RMS configurations (Xu and Liang 2005; Xu and Liang 2006; Ye and Liang 2006). As previously stated, in this work, two propositions are made to concurrently optimize:

- 1) The product configuration with the process planning in an RMS;
- 2) The product configuration with the process planning and layout configuration in an RMS.

Both problems are integers and then non-convex and non-smooth. These problems are characterized to have many feasible solutions, with a solution space composed of several regions containing multiple local optimal points. When problems with mid-large instances are addressed, it becomes impossible to enumerate all the possible solutions to get the best one.

GA belongs to the Evolutionary Algorithms (EA) class and is a population-based algorithm capable of executing random parallel searches from a population of points, avoiding being trapped in local optimum solutions. Hence, GA presents a good ability to solve various optimization problems, especially non-smooth and non-convex ones, which cannot be solved using traditional optimization algorithms in a reasonable computation time when higher instances are addressed (Gandomi et al. 2013).

Besides that, GA allows dealing with various parameters and groups of encoded string simultaneously (Yang 2010). In other words, the global optimization problem can be encoded in a simple or 2D string using different types of codes (real, binary, or integer values) according to the need. This allows GA to solve multiple sub-problems simultaneously that are interconnected, such as the configuration of product and RMS, as proposed in this work.

Considering the advantages of the GA and its robustness to solve problems close to the ones addressed in this work, two GA-based approaches were applied to solve concurrent

optimization of product configuration with the process planning or/and layout configuration in an RMS.

### 3.1.1. Genetic algorithm

Initially proposed by (Holland 1975), GA is a nature-inspired algorithm based on biological evolution over generations to solve complex optimization problems (Yang 2010; Gandomi et al. 2013). This means that GA combines surviving fittest individuals (strings/chromosomes) from each generation with randomized information in a structured way, allowing an effective search through the solution space (Goldberg 1989).

The main components of GA are described as following (Srinivas and Patnaik 1994):

- Genetic representation of encoded solutions called strings or individuals or chromosomes;
- Population comprising a set of encoded solutions;
- Fitness function to evaluate the optimality of each string;
- Genetic operators which are responsible for generating the next generation from the existing population.

The well-structured evolution process of GA is mainly due to the genetic operators, which are described below:

- I. Selection: is used to choose the “parents” that will recombine to create offsprings for the following generations. Besides the different selection mechanisms, tournament selection stands out as one of the mainly used since it has proven to be more efficient and prone to premature convergence (Goldberg and Deb 1991). Tournament selection takes only the fittest chromosome from a set of strings randomly selected from the current population to be placed in the mating population (Reed et al. 2000). The number of strings to be randomly selected from the current population is related to the *tournament size* parameter.
- II. Crossover: randomly takes two parent strings to create a new offspring. *Crossover rate* corresponds to the parent population percentage that will undergo a crossover operation. The main advantage of GA arises from this operator since through a random and structured exchange of genetic material within different chromosomes,

the crossover allows to combine good solutions for creating still better solutions (Srinivas and Patnaik 1994). High values of crossover rate can imply good individuals being discarded, while low levels of crossover rate considerably increase the parents' priority, leading to a stagnation of the optimization search (Ortiz et al. 2004).

III. Mutation: randomly changes an offspring to prevent solutions of a specific population from falling into a local optimum. The mutation rate is the probability of mutating a chromosome. High mutation rates under high selection pressure can imply an essentially random evolution, leading to the extinction of the whole population, while the low ones reduce the evolution efficiency (Yang 2010).

As GA is a gradient-free method, it has a good ability to solve different types of optimization problems, finding good solutions for complex problems in a reasonable computational time (Yang 2010). Two features, named exploitation and exploration, are essential for achieving that. While the exploitation phase ensures the search of the current best solution and selects the best candidates, the exploration phase is responsible for exploring the search space efficiently (Gandomi et al. 2013).

Both phases together have an essential role in the GA performance, which is closely dependent on finding a good balance between them. Exploitation and exploration directly rely on how the GA parameters are adjusted, which includes the crossover and mutation rate, the selection type, tournament size, the number of generations, and the parent/offspring ratio (Gandomi et al. 2013).

Different versions of GA arose from the initial GA proposed by (Holland 1975), including messy genetic algorithms (mGA) (Goldberg et al. 1989), nested genetic algorithm (Liu et al. 2017) and non-dominated sorting genetic algorithm (Deb and Jain 2014). However, they are all based on the standard genetic algorithm's basic structure, which is described by the algorithm below (Malhotra et al. 2011).

---

**Algorithm 1: Standard Genetic Algorithm**

---

Step 1	[Start] Generate a random population of chromosomes (suitable solutions for the problem)
Step 2	[Fitness] Evaluate the fitness of each chromosome in the population
Step 3	[New population] Create a new population by repeating the following steps until the new population is complete:

	I.	[Selection] Select two parent chromosomes from a population according to their fitness. Better the fitness, the higher the chance to be selected to be the parent
	II.	[Crossover] Based on the crossover probability, cross parents to form new offsprings (children). If no crossover was performed, offspring is exactly the parents' copy
	III.	[Mutation] Depending on the mutation probability, mutate new offspring at each locus
	IV.	[Accepting] Place new offspring in the new population
Step 4	[Replace]	Use the new generated population for a next run of the algorithm
Step 5	[Test]	If the end condition is satisfied, stop the evolution process and return the best solution from the current population
Step 6		Go to step 2

---

### 3.1.2. GA-based solution approaches

In this work, both GA-based approaches differ from the Standard GA, because they were based on the application of different chromosome coding types and the use of multiple and specific genetic operators. Further, for the second GA-based approach, denominated messy GA, the variable-length strings were also used. Sections 3.1.2.2 and 3.1.2.3 will present more details about both methods.

Messy GA is a specific kind of GA that combines variable-length strings with messy operators, overcoming the limits stated by the neatness of standard GA, and being capable of solving more complex optimization problems effectively (Goldberg et al. 1989; Goldberg et al. 1990). Using more messy operators and chromosomes allows the GA to exploit and explore the solution space with much higher performance than standard algorithms having fixed random coding and single operators (Goldberg et al. 1989).

Instead of fixing specific crossover and mutation rates, some researchers have proposed using adaptive genetic operators that can change their rates according to the evolution process, allowing a balance between exploitation and exploration phases on the search for the solution (Srinivas and Patnaik 1994; Chaves et al. 2018). Adaptive methods have proven their ability to improve GA's convergence rate, preventing the algorithm from getting stuck at local optima (Srinivas and Patnaik 1994). Further, they can help solve multimodal problems (i.e., problems having multiple satisfactory solutions) with large solution spaces. In this work, one GA-based approach uses adaptive crossover and mutation rates, while the other uses fixed probability values. For the latter, the optimal rates are stated through a 2-full factorial design method that will be further discussed in Chapter 5.

Besides the types of strings coding and genetic operators types and rates, another parameter related to the GA effectiveness is the selection mechanism. The reproduction selection and the deletion of individuals are made in the selection pool. An essential parameter



of the selection is the generation gap, which refers to the quantity of overlap between parents and offsprings (Sarma and Jong 1995). Historically, the standard GA is mainly based on the non-overlapping populations, meaning that the offspring at each generation replace the entire current population present in the selection pool. On the other hand, the overlapping models present a selection pool composed of the parents and offsprings; therefore, both are continually competing for survival at each generation (Sarma and Jong 1995).

Since all parents are replaced in non-overlapping strategies, there is no guarantee that good individuals with high fitness will survive in the next generation (Sarma and Jong 1995). That can lead to a loss of effectiveness in the exploitation and exploration, hampering an appropriate convergence of the algorithm, especially in complex problems with large solution spaces. When dealing with problems presenting large solution spaces containing many solutions, the overlapping models are more appropriate because they ensure that most current individuals presenting high fitness will be selected for the next generation (Sarma and Jong 1995). Considering the complexity of the problems presented in this work, the overlapping mechanism was used in both GA-based approaches.

Although both approaches presented in this work are based on GA, they present some specificities. Before going through each method's details, Section 3.1.2.1 will present the mechanisms used in each approach to code the strings and handling, which involves the genetic operators. Sections 3.1.2.2 and 3.1.2.3 will detail the GA-based approaches used to solve both mathematical problems previously presented.

### **3.1.2.1. Chromosome encoding and decoding and constraints handling**

As summarized in Table 9, the first problem (Proposition 1) to be solved in this work is a ILP that only englobes the product configuration and the process planning. Besides the product configuration and process planning, the second problem (Proposition 2), a NLIP model, also integrates decisions related to the layout configuration. Considering that, all descriptions regarding the encoding mechanisms and genetic operators for layout configuration only refer to the methods used to solve the NLIP. The GA-based approaches used to solve the ILP will only be encoded to represent product configuration and process planning issues.

The process planning englobes two decisions: the operations' sequence and their assignment to machine-configurations. Besides them, there is also the decision associated with the product configuration and the layout configuration in case of NLIP problem. Hence, the chromosome is encoded in three and four parts for propositions 1 and 2, respectively.

Table 9. Set of sub-problems addressed by each of the mathematical models proposed in this work.

Sub-problem	Description	Addressed by proposition:
Product configuration	Selection of modules compounding the product variant and identification of operations required by them.	1 and 2
Process planning	i) Sequencing of required operations and ii) their assignment to machine-configurations.	1 and 2
Layout configuration	Definition of which machines will assume which layout locations.	2

Figure 11 illustrates how the GA chromosomes were encoded in this work. Part 1 corresponds to the module instances selection for configuring the product and is encoded with real values. Operations' sequence is encoded in part 2 using integer values. Machine-configuration selection is encoded in real values and represented in part 3. Finally, layout configuration (only for NLIP problem) is encoded in integer values (part 4).

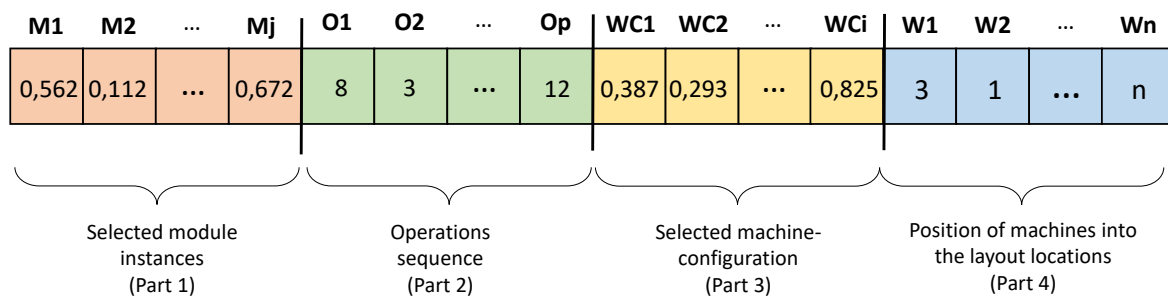


Figure 11. GA encoding.

Just as important as encoding the chromosome is decoding it in order to find the correspondent values that represent the searched output. The decoding process of each chromosome part is described in Figure 12.

The integer values of part 2 (operations) correspond to the operations' values and the position they appear corresponds to their position in the operations' sequence. In Figure 11 (green part), for example, the first gene of operations part is coded with the value 8, meaning that operation 8 appears in the first position of the operations' sequence. Part 4 is also encoded with integer values but here the value appearing in a gene corresponds to the layout location in which the corresponding machine is placed. Machine 1 always corresponds to gene 1 of part

4, machine 2 to gene 2, and so on. Therefore, for the example presented in Figure 12bFigure 11, machine 1 is placed into location 3 and machine 2 into location 1.

For part 1 (module instances) it is assumed that the first gene always corresponds to an instance of module 1, the second gene to an instance of module 2, and so on. Therefore, to find the module instance corresponding to the real value from the gene, this value is multiplied by the total number of available instances from the related module and rounded to the closer integer value. In the example illustrated in Figure 12a, module 1 has 4 available instances; then, 4 is multiplied by 0.562 and the result is rounded to 2. Therefore, the value 0.562 in gene 1 corresponds to instance 2 of module 1.

For the machine-configuration's part (part 3), the procedure is somewhat different because part 3 is dependent on part 2. The first gene in machine-configuration's part corresponds to the machine selected to do the operation in the first gene of part 2 (operations). The coded value of the gene in part 3 is multiplied by the number of machine-configurations capable of accomplishing the corresponding operation. In the example illustrated in Figure 12c the value 0.387 (from part 3) is multiplied by 3 since there are three machine-configurations capable of performing the operation 8. This multiplication resulted in the value 1, which corresponds to machine 1 with configuration 2.

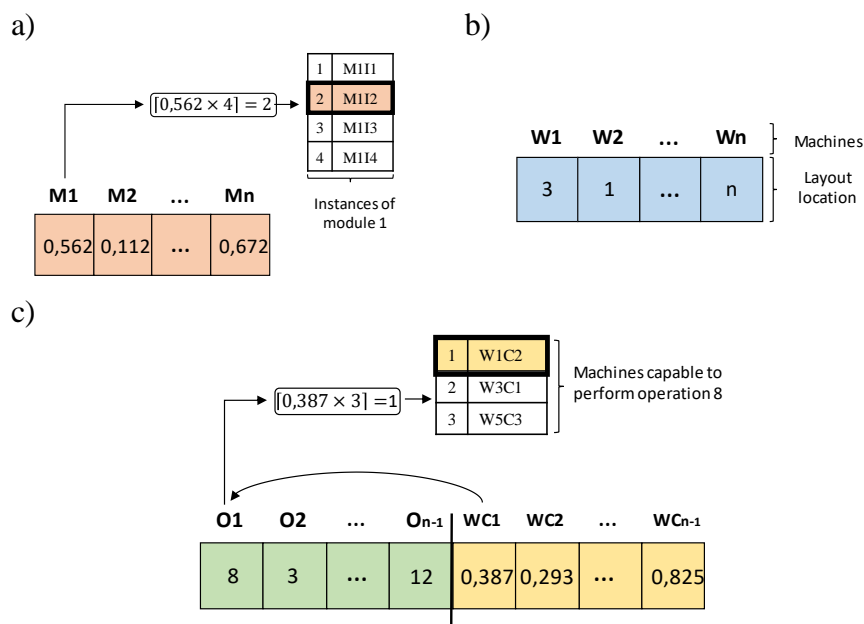


Figure 12. Decoding the chromosome: a) Part 1: module instance selection and b) Part 2 and 3: operations sequence and selection of machine-configuration for each operation and c) Part 4: selection of layout locations for each machine.

The strategy used in part 3 ensures that the machine-configuration selected will always be able to perform the corresponding operation. Similarly, part 1 is encoded in real values to ensure that the module instances selected will always belong to the corresponding module. Using real encoded values mixed to integer ones is a good strategy to deal with multiple sub-problems simultaneously while ensuring that hard constraints will be respected. Encoding the same problem only with binary encoding as generally used by the standard GA would hamper the representation of all these sub-problems into the string, hence hampering to solve the two proposed problems (ILP and NLIP) by using GA-based approaches.

There are different genetic operators adopted for each coded part of the chromosome. The genetic operators used to mutate and cross individuals in the GA-based approaches proposed in this work are presented as following:

*Crossover:*

- One-point crossover selects a random value within the length of each of the parents' strings, and the genetic materials of both parents are permuted to get new offsprings (Figure 13 – on the left). The resulting individuals will respectively have the length of each other. This crossover was applied on the real coded parts.
  
- Heuristic one-point crossover refers to the one-point crossover followed by a heuristic operation, in which both individuals' gene values are checked and filtered to ensure that the final offsprings will not have repeated values (Figure 13 – on the right). This crossover was applied to the parts coded with integer values because each operation can be performed once, and each machine type can appear in a layout position once.

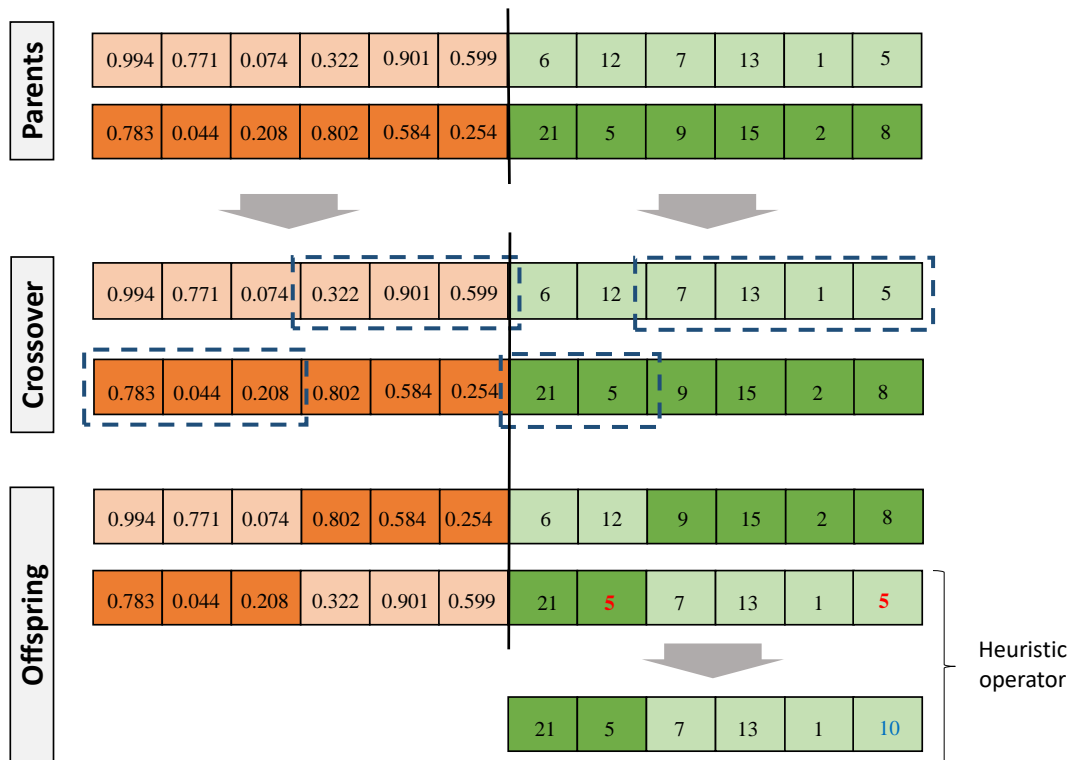


Figure 13. Standard one-point crossover for real-coded strings (left) and heuristic one-point crossover for integer-coded values.

*Mutation:*

Random float mutation: applied for real-coded values, this mutation randomly selects a gene from the chromosome and replaces it with a random value within the interval (0,1), as illustrated in Figure 14.

Swap mutation: corresponds to the swapping of two genes within the chromosome. This mutation was applied in integer-coded parts with fixed lengths (Figure 14).

Modified random resetting mutation: this mutation was applied in integer-coded parts with variable lengths. This mutation presents three different mechanisms that are applied under certain conditions:

- i) If the string contains invalid elements (invalid gene values), an element from the list of valid values is randomly chosen to replace the invalid element in the chromosome;
- ii) If the string contains invalid elements, but there is no valid element to be added, the invalid element is removed from the string;
- iii) If there is no invalid element in the chromosome, but there is still valid genes to be added, a valid gene is randomly chosen from the set of valid candidate values;

- iv) If there is no invalid element to remove or no valid element to add, a swap mutation is carried out.

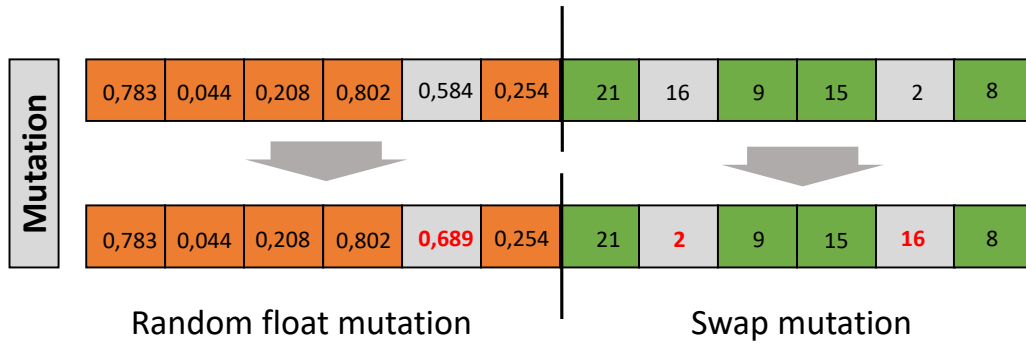


Figure 14. Illustration of the random float and swap mutations.

Using real and integer values into the same string and applying customized genetic operators do not ensure that all of the hard constraints will be respected. This is why penalty functions were applied to handle other constraints, including that related to the operations precedence, invalid operations, module instance compatibility, customer requirements satisfaction etc. In order to accurately manage these penalties, the fitness function was normalized to values in the (0,1] range, as presented in Eq. (56).

$$z_{norm} = \frac{(1 - 0) * (z - c)}{d - c} + 0 = \frac{(z - c)}{d - c} \quad (56)$$

Where:

$c, d$  represent the minimal and maximal values of the fitness values before normalization ( $z$ ).

Therefore, the fitness function minimized by GA corresponds to the sum of the normalized fitness function ( $z_{norm}$ ) with the penalties, as represented in Eq. (57).

$$f = z_{norm} + \sum_n \Delta_n \omega_n \quad (57)$$

Where:

$\Delta_n \begin{cases} = 1, \text{if constraint } n \text{ is violated} \\ = 0, \text{otherwise} \end{cases}$   
 $\omega_n$  : penalty weight for constraint  $n$

### 3.1.2.2. Exhaustive search + GA

As the name suggests, this approach is based on a combination of an exhaustive search algorithm (ESA) and GA. The exhaustive search algorithm is used in the first phase (of ILP and NLIP models) to filter all module instance combinations (i.e., product variants) capable of satisfying the set of selected functional requirements while respecting module instances' compatibility constraints. The pseudocode of the exhaustive search algorithm used here is detailed in Algorithm 2.

---

**Algorithm 2:** Pseudocode of the exhaustive search algorithm (ESA):

---

*Begin*

*input customer\_requirements*

*input module\_instances*

*input matrix\_module\_inst\_compatibility*

*input matrix\_requirements*

**function** *remove\_val(List, Value):*

**while** *Value in List:*

*remove Value from List*

**function** *generate\_all\_combs\_with\_m1(List):*

*generate all combinations from elements in List*

**for** *comb in combinations:*

*remove\_val(comb, None)*

*selected\_combs = [ ]*

**for** *comb in combinations:*

**if** *comb has module 1:*

*selected\_combs.append(comb)*

**return** *selected\_combs*

**function** *filter\_with\_functions(combinations, customer\_requirements, matrix\_module\_inst\_compatibility, matrix\_requirements):*

**for** *comb in combinations:*

**if** *comb does not satisfy customer\_requirements according to matrix\_requirements:*

*remove comb from combinations*

**if** *comb contains at least two module instances that are not compatible at*

*matrix\_module\_inst\_compatibility:*

*remove comb from combinations*

**return** *combinations*

*product\_configurations = filter\_with\_functions(*

*generate\_all\_combs\_with\_m1(module\_instances),*

*customer\_requirements,*

*matrix\_module\_inst\_compatibility,*

*matrix\_requirements*

)

*End*

---

While ESA is applied in the first step, in the second step, GA is applied to solve the process planning (for ILP) or process planning and layout configuration (for NLIP). Since ESA does the module instances selection, in this approach, the GA's string does not include part 1 of the chromosome described in Section 3.1.2.1. Further, as previously explained, for the ILP model, the RMS configuration includes only the process planning, meaning that only parts 2 and 3 are considered. For NLIP, on the other hand, the RMS configuration also includes layout configuration, and then the string parts 2 to 4 are considered in the chromosome coding.

The ESA-GA presents a fixed-length chromosome because once ESA selects the product variant, the operations to be performed (i.e., operations required by module instances) are known. Then, they must be just sequenced and assigned to machine-configurations, meaning that the chromosome length will be dependent on the number of required operations, but it will never change within the same product variant.

Considering that, the GA conducts an optimization of process planning only (Proposition 1) or process planning and layout configuration (Proposition 2) for each product variant individually. Therefore, the higher the number of candidate product variants, the higher the computation time, which can be a limitation of this method to deal with problems containing high quantities of product variant candidates. In contrast, it can be effective in finding optimal solutions because the solution space at the GA level is reduced, and there are fewer decisions to be made since there is not the product configuration part.

Figure 15 presents the framework of the ESA+GA for solving the ILP and NLIP problems proposed in this work. The procedure to solve both problems is quite similar, except by the layout configuration (part 4), which is only included in the NLIP problem. Therefore, costs of changing layout configuration are only included in the fitness function of NLIP; all other costs (i.e., raw material, operations, material handling, and machine reconfiguration) are included in both problems.



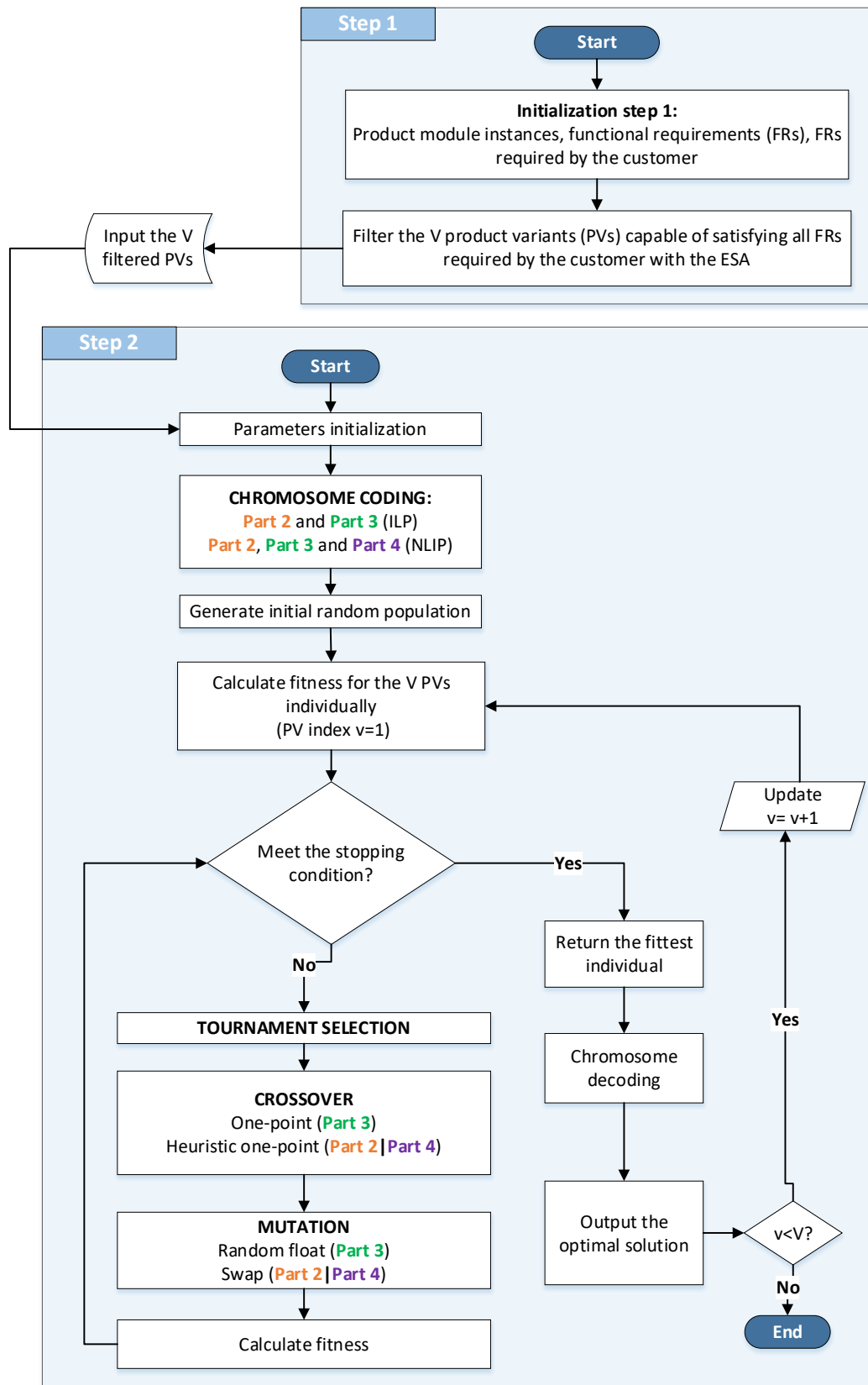


Figure 15. The framework of the ESA+GA approach for solving the ILP (Proposition 1) and NLIP (Proposition 2) problems.

### 3.1.2.3. Messy GA

The second proposition is a kind of messy GA (mGA) that presents variable-length and adaptive genetic operators. This method is proposed to overcome the limitations of the ESA-GA regarding the number of product variant candidates. As previously stated, the ESA-GA can be time-consuming for problems containing a large quantity of product variant candidates for a given set of selected FRs, because decisions at the RMS level must be taken for each product variant candidate separately. The proposed mGA, on the other hand, includes all decisions simultaneously since product configuration decision is included in the chromosome.

Including the product configuration decision in the GA chromosome considerably increases the complexity of this problem. Besides sequencing operations and assigning them to given machine-configurations and configure the layout, it is necessary to select the module instances of the product variant and then identify which operations are required to produce the selected product variant.

Each product variant is composed of various module instances, requiring different quantities and types of operations. Considering that, the chromosome length must be variable to ensure that only the operations required by the selected product variant will be manufactured, meaning that sometimes it must be short and other times longer to accommodate all required operations adequately. This is why here the chromosome has variable-lengths for operations' sequence (part 2) and machine-configurations selection (part 3), because the latter is directly dependent on part 2, as previously explained. Nevertheless, part 1 and part 4 (in the NLIP model) keep fixed lengths.

A large number of product variant candidates, together with the numerous possible operations and the uncountable ways of sequencing them, can considerably enlarge the solution space, increasing the probability of the algorithm being trapped in inadequate solutions. A key step to widely explore the search space is the initialization (Gandomi et al. 2013). There are different methods to initialize the population, such as random generation and heuristic initialization (Gandomi et al. 2013). Heuristic initialization refers to seed the initial population with good individuals. Random initialization means initializing the population with entirely random individuals.

Suppose a population is initialized only with good individuals. In that case, it tends to produce solutions very similar or even identical to each other, considerably reducing the diversity, which is not desirable in evolutionary algorithms (Burke et al. 1998). In problems presenting a huge search space, initializing the population with completely random individuals could lead the population to be trapped into local optima or even invalid solutions. The heuristic initialization suggests a good balance between random and good individuals to keep enough diversity while ensuring that individuals with good fitness will, in some way, guide the evolution process in order to improve the average fitness.

When using heuristic initialization, some initial fitness values are close to the optimal ones, saving a high amount of time. Further, the initial population presenting high fitness allows the algorithm to focus on the “fine-tuning” of the solutions, improving the search for optimal solutions (Burke et al. 1998). Complementary to that, Burke et al. (1998) also state that evolutionary algorithms may perform much better when some knowledge of the problem is used to help on the optimization process.

Considering the advantages of the heuristic initialization and knowing that the problems addressed by this work are multimodal, including different subproblems, which are known to be hard to solve, the heuristic initialization was used in the mGA. A pre-processing (Figure 16) based on the ESA and constraint propagation (Algorithm 3) was carried out to reduce variable decisions’ domains and get good individuals to subsequently being used to feed the initial population.

The domain reduction was applied within the product configuration (part 1) and operations sequence (part 2) of the algorithm. The objective of this pre-processing was to select good individuals according to the following criteria:

- 1) Part 1: individuals presenting product variants satisfying all FRs required by the customer while being composed only by compatible module instances;
- 2) Part 2: individuals presenting only the operations required by the corresponding product variants in a valid sequence (i.e., respecting the precedence constraint).

There were no worries concerning part 3 because it was coded in such a way that the machine-configurations selected will always be capable of fulfilling the corresponding operations. Part 4 was not considered because there is no constraint regarding the machines’ placement into the layout locations. Therefore, all layout configurations are valid, but some of them will be preferable depending on the current layout configuration.

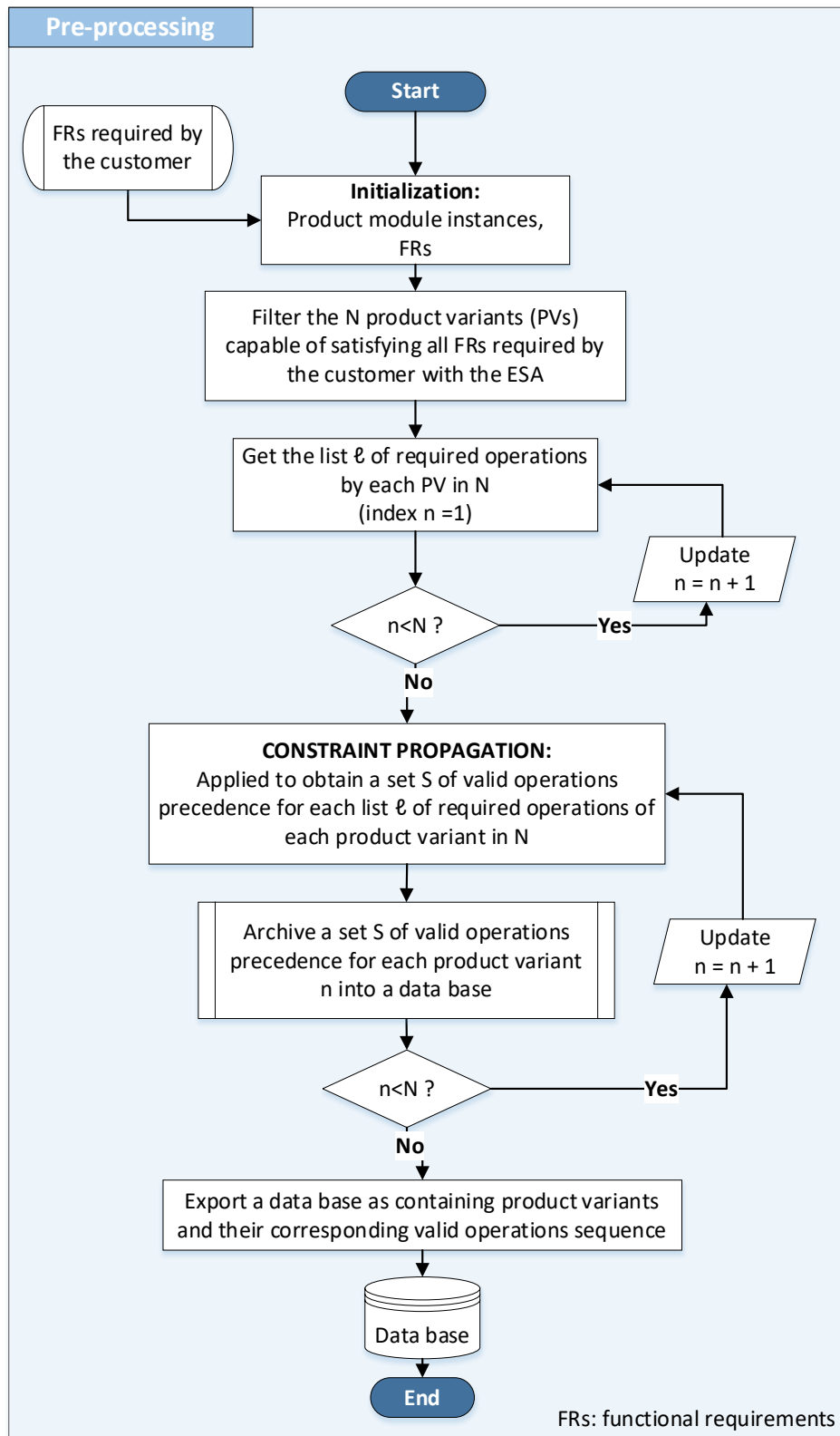


Figure 16. The framework of the pre-processing method applied to get good individuals to inject into the initial population of mGA.

---

**Algorithm 3:** Pseudocode of the constraint propagation method

---

*Begin*

$I$  is the set of operations required by a product variant  $v$ , where  $v = 1, 2, \dots, |V|$

Let the variable  $x_{ij} = 1$ , when the operation  $i$  is processed in the process sequence  $j$  and 0, otherwise.  
Where  $i = 1, 2, \dots, |I|$ , and  $j = 1, 2, \dots, |J|$ , where  $|I| = |J|$ .

Let  $P_{iiv'} = 1$ , when operation  $i'$  must be preceded by operation  $i$ . 0, otherwise.

for  $v$  in  $|V|$ :

for  $j$  in  $J$ :  
 $\sum_i x_{ij} = 1$

for  $i$  in  $I$ :  
 $\sum_j x_{ij} = 1$

$\sum_{ij} x_{ij} = |I|$

for  $x$  and  $x'$  in *required\_operations*:

for  $j'$  in  $J$ :  
 $\sum_j^{j'-1} x_{ij} \geq P_{iiv'} \times x_{ij'}$

**Solve**

**return** ( $v$ :  $S$ ) // Where  $S$  is the set of operations sequence obtained by the constraint propagation

*End*

---

The number of good individuals to be generated depends on the number of random individuals and must always be less than the latter to ensure enough diversity. The proportion between good individuals and random ones is stated in the algorithm's initialization. Figure 17 presents the framework of the mGA, showing that the initial population always contains good individuals and randomly generated ones.

In order to ensure that the good individuals injected in the first generation would not be lost over the generations, an elitist mechanism was applied in parallel to the tournament selection to ensure that 10% of the best individuals, among the random and goods ones, would be selected to the next generation. In the following generations, the elitist mechanism was also applied in parallel to the tournament selection, but instead of 10%, only the best individual (best fitness) was taken at each generation. This elitism mechanism's objective was to ensure that good individuals would not be lost over the generations.

In mGA the probabilities of crossover and mutation were variable according to the fitness function ( $fit\_func$ ) and standard deviation ( $fit\_std$ ) of the population fitness. The crossover and mutation probabilities started with the following values  $p_c=0.7$  and  $p_m=0.3$ , which were defined based on preliminary analyses. When  $fit\_std \geq 0$  and  $fit\_func \geq target\_value$ , for generations  $\geq 0.75 * NGEN$ , where  $NGEN$  corresponds to the total number of generations, the  $p_c$  increased to 0.9 and  $p_m$  decreased to 0.1. The  $target\_value$  is problem-dependent and must be defined based on preliminary analyses or knowledge about the problem to be optimized.

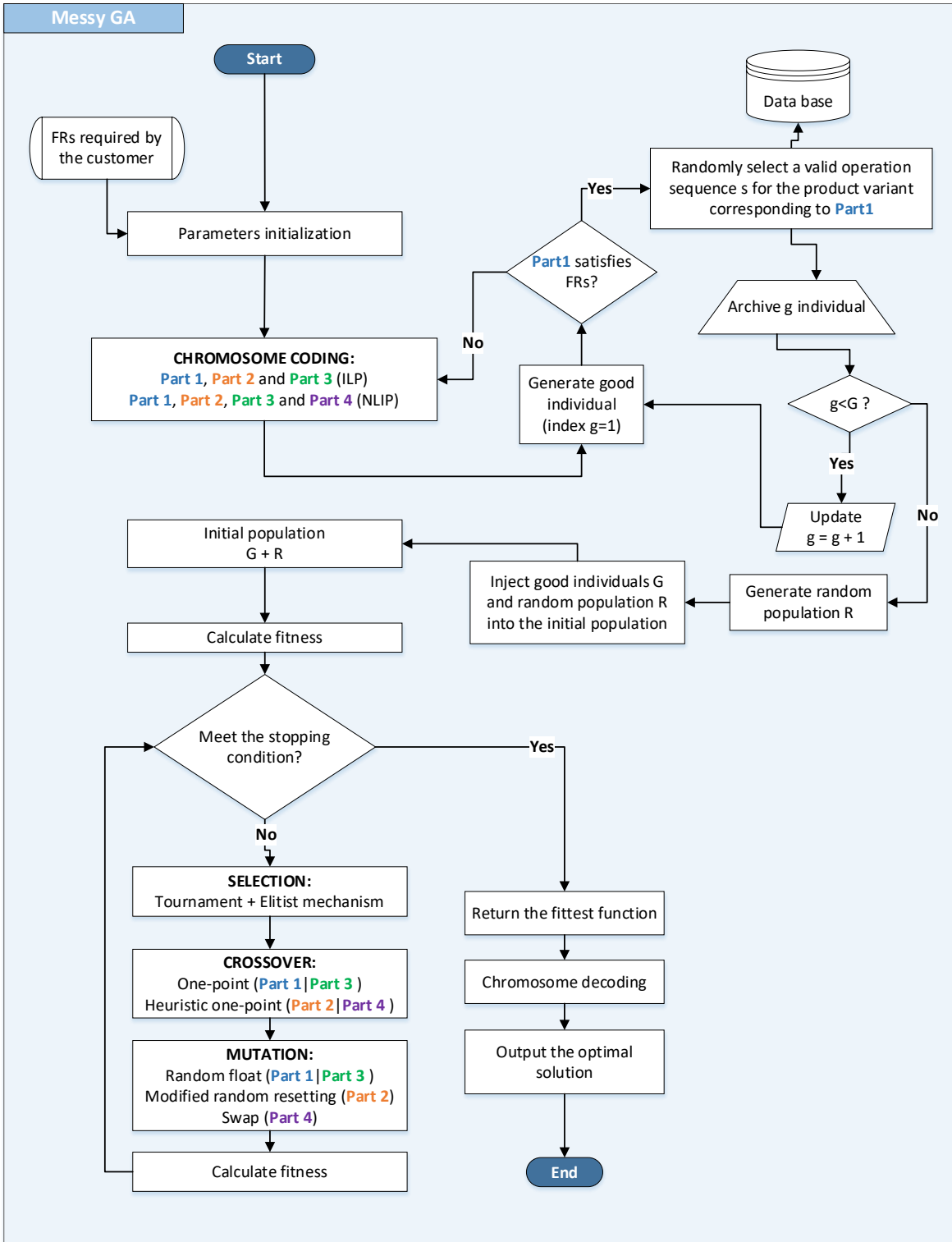


Figure 17. The framework of the messy GA.

### 3.2. Final considerations from this chapter

This chapter introduced two solution approaches based on GA to solve the mathematical models proposed in Chapter 2. The literature review has shown that GA-based approaches have effectively solved configuration problems for modular products and RMS, standing out as one of the main techniques used to solve the configuration of a modular product or RMS separately and for solving both concurrently. Considering the advantages of GA and its robustness to solve problems close to those addressed in this work, both approaches were based on this algorithm.

The chromosomes were encoded with real and integer values in both approaches, contributing to handling some constraints. The operations sequencing could be easily managed with their values being encoded into integer values. Likewise, integer-encoding values for machine positions favoured machines' position changeovers. In both approaches, machine-configuration was coded with real values, which corresponded to a machine-configuration capable of fulfilling the corresponding operation.

Only mGA included product configuration decision into the chromosome, and it was also real-coded following the same principle of machine-configuration encoding. Each gene of the product configuration part corresponded to a variant of the corresponding module. One limitation of this method is that the product always needs to include a module instance of each of the existing modules, meaning that optional modules are not allowed. ESA-GA, on the other hand, can handle product configurations presenting optional modules.

The ESA-GA consists of a hybrid approach in which GA is coupled with an exhaustive search algorithm. The ESA is responsible to identify all feasible product variants capable of satisfying the set of FRs. GA comes to optimize the process planning (ILP and NLIP) and layout configuration (NLIP) of each product variant individually. The advantage of separating this problem into two steps is that once module instances are selected, the required operations are known. Hence, it becomes possible to optimize the process planning (in the ILP and NLIP problems) and the layout configuration (NLIP) with no need to varyate the chromosome length for inserting or removing operations.

A drawback of the ESA-GA approach is that it is directly dependent on the number of product variant candidates. The higher the number of candidates, the higher the computation time. Hence, ESA-GA becomes very time-consuming for problems considering large instances. The mGA was proposed to overcome this ESA-GA's limit. In



this approach, all decisions are made by GA. Therefore, the number of product variant candidates will not significantly affect the computation time.

The solution space addressed by mGA is considerably higher than that of ESA-GA at the evolutionary process, especially for high instances problems, because mGA addresses the process planning and layout configuration (for NLIP) for all possible product variants; while ESA-GA addresses them for each product variant candidate individually at the GA step. Including product configuration decisions into the chromosome forced mGA to have a variable length for accommodating the different operations required by each product variant, which is more challenging to handle than fixed-length chromosome algorithms. Besides that, the higher solution space increases the possibility of the algorithm being trapped into local optima.

Chapter 4 will present the validation of both solution approaches by comparing the results obtained by them with those obtained by a deterministic approach for solving an example with small instances. Chapter 5 will further discuss each method's performance through two illustrative case studies presenting mid-size instances. At the end of Chapter 5, a global comparison about each method's performance on solving both problems is carried out and other metaheuristics are suggested as alternatives to the methods proposed in this work.

## CHAPTER 4 – PROPOSITIONS’ VALIDATION

### 4.1. Methods to validate the mathematical models

A deterministic approach was used in order to validate the mathematical models. This approach was based on the combination of an ESA (presented in Algorithm 2) and CPLEX. The exhaustive search algorithm is used in the first phase (of ILP and NLIP models) to filter all module instances combinations (i.e. product variants) capable of satisfying the set of PFs required by the customer, while respecting module instances compatibility constraints.

CPLEX is a commercial software based on the combination of branch and bound algorithm with cutting plane method (Mitchell 2008). CPLEX is capable of solving ILP problems but unable to find the optimum solution for nonlinear programming ones. As previously presented, the first mathematical model was modelled as an ILP while the second one as an NLIP.

For the mathematical model for fixed layouts (ILP), CPLEX was used in the second step to find the optimal process planning for all product variants individually (Figure 18). The optimal solutions found for the product variants were ranked (i.e., process planning with the minimal cost), and then the global optimal solution corresponding to the minimal overall manufacturing cost – including costs of product variant and process planning was selected.

In the second step of the NLIP model, machines were permuted among all available layout positions in order to identify the number of available layout configurations. The process planning of each possible layout (with fixed machines) was individually optimized with the aid of CPLEX (Figure 19). Fixing machines into layout locations linearizes the problem, allowing CPLEX to solve it. The optimal solution found for each layout (optimal process planning and product variant) was ranked in the third step and then the optimal global solution with the minimal overall manufacturing cost (i.e. sum of costs of raw material of product variant, process planning and layout configuration) was selected.

The layout configuration was modelled as a QAP, meaning that the number of machines is equal to the number of layout locations, which is ensured by the constraint Eq. (49). Considering that, the number of possible layouts corresponds to the permutations of machines and layout locations. Therefore, a problem with 4 machines corresponds to  $4! = 24$  layouts, 5 machines  $5! = 120$  layouts and so on.

With that in mind, in the NLIP problem, CPLEX needs to iterate over each layout. The higher the number of layouts, the higher the number of iterations, and subsequently the higher the computation time required. For small-size NLIP problems, the exhaustive search + CPLEX

approach can be relatively efficient. However, for mid/large-sized problems, this approach becomes inefficient.

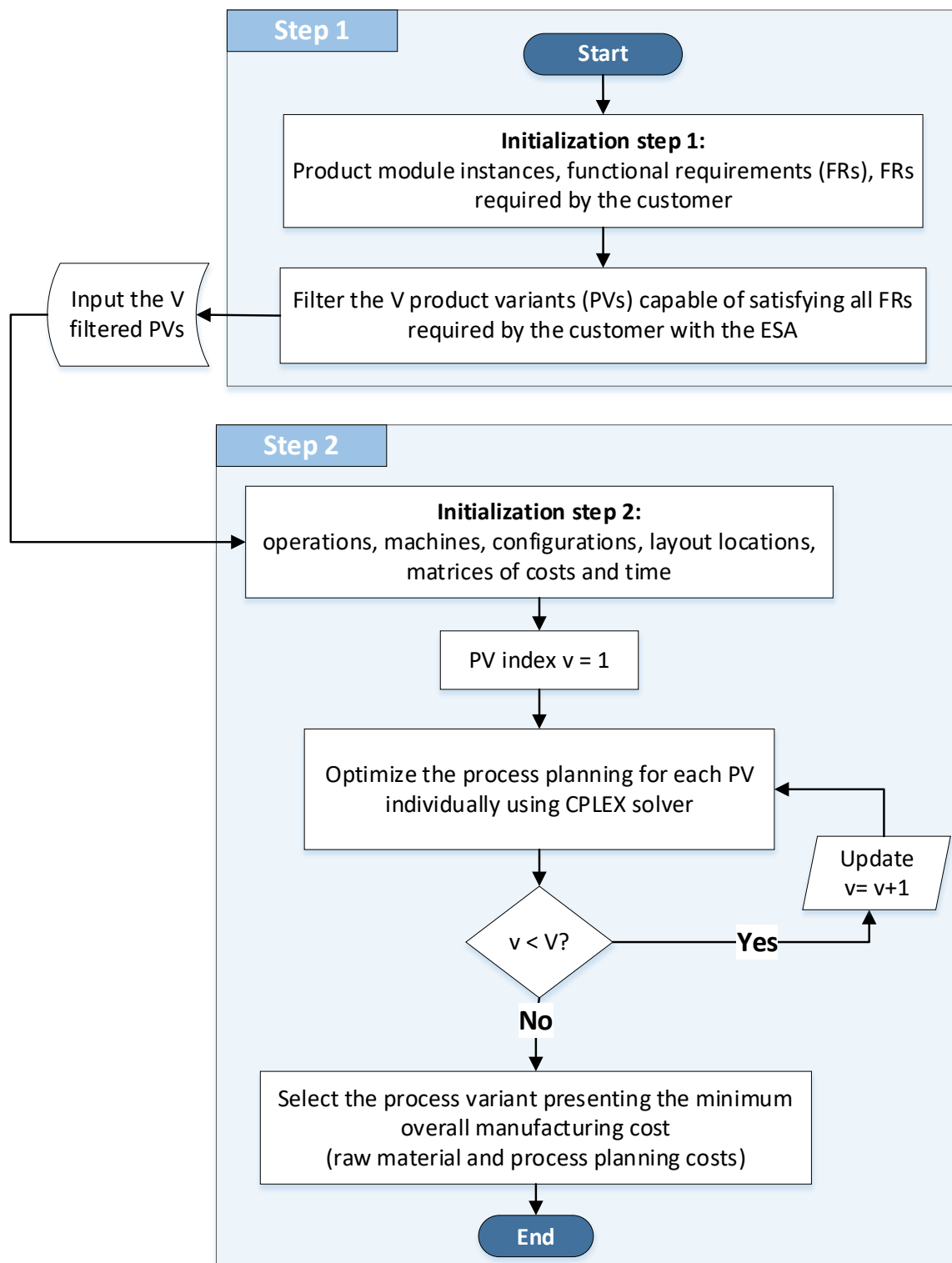


Figure 18. Framework of the exhaustive search and CPLEX for solving the ILP problem.

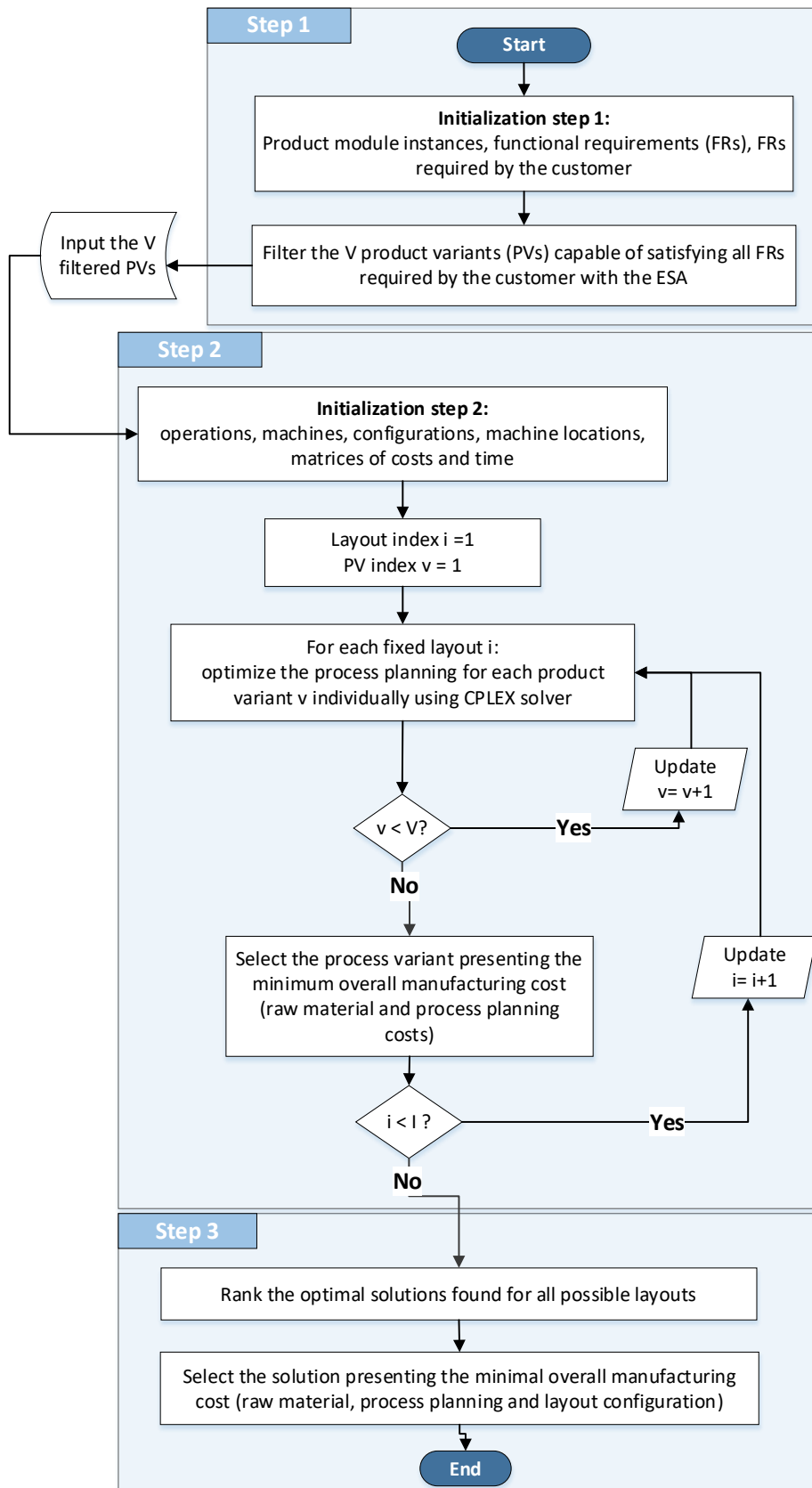


Figure 19. Framework of the exhaustive search and CPLEX for solving the NLIP problem.

#### 4.2. Mathematical models validation

In order to validate the mathematical models previously presented in Chapter 3, a small-size example containing 11 available product functions, 4 modules (total of 10 module instances), 16 operations and 4 machines (total of 14 machine-configurations) was implemented. Further details concerning this example are presented in Appendix A.

Both mathematical models were validated using the approaches detailed in Section 4.1 through the aid of CPLEX solver, which was implemented in Python 3.7 language. DCOplex Python API was used for modeling the mathematical programming problem as well as to import CPLEX solver. The calculations were carried out in a laptop computer powered by an Intel core i7-7600U CPU(2.80 GHz) and 16 GB of RAM.

Table 10 summarizes the main inputs used in both mathematical models. For both examples, the same required product functions and initial machines' configuration were chosen. Since fixed layout does not address the layout configuration, there is not an initial layout configuration to be considered. Instead, there is a fixed configuration, meaning that only the layout configuration presented in Table 10 will be considered for all examples. While W1C4 means configuration 4 of machine 1, W1:1 means that machine 1 is placed into the layout position 1. As previously stated, the distances between layout positions/locations are known.

Table 10. Input parameters used in the example to validate both mathematical models.

Mathematical model	Required product functions	Initial machines' configurations	Initial (or fixed) layout configuration
<b>1. ILP</b>	2, 6, 7, 10	W1C4, W2C2, W3C2, W4C5	W1: 1, W2: 2, W3: 3, W4: 4
<b>2. NLIP</b>	2, 6, 7, 10	W1C4, W2C2, W3C2, W4C5	W1: 1, W2: 2, W3: 3, W4: 4

Table 11 presents the results obtained by Exhaustive search-CPLEX for solving linear (fixed layout) and nonlinear (changeable layout) problems. In both cases, the results obtained were the same in terms of selected module instances, process plan, and total cost. This can be explained by the fact that the optimal solution for changeable layout corresponds to the layout with fixed configuration, which is presented in Table 10. Since the cost rate multiplied by the time of displacing machine is, on average, much higher than the cost per time of changing machine

configuration or transporting material between two machines, the cheapest cost is that in which machines are not displaced.

Although both problems, fixed and changeable layout, obtained the same optimal results, the computation time required by the nonlinear (changeable) problem was about 30 times higher than that required for the linear one (fixed). Since CPLEX can solve only linear problems, it is unable to directly solve the problem of the changeable layout. This is why an additional solution step was used to enumerate all possible layout configurations, and then the process plan was calculated individually for each of them. There were 4 machines in this problem, meaning that there are 24 possible layouts ( $4! = 24$ ), which justifies the longer computation time.

Table 11. Results obtained with Exhaustive search + CPLEX for both mathematical problems: 1) ILP and 2) NLIP.

Exhaustive search-CPLEX				
1. ILP	<b>Module instances</b>	<b>Total cost</b>		<b>Computation time</b>
	M1I2, M2I1, M3I2, M4I2	144.6€		122,2 s
	Process plan	1:OP1 (W22) →2: OP3 (W22)→ 3: OP6 (W14)→ 4: OP8 (W14)→5: OP7 (W21)→6: OP10 (W21)→ 7: OP14 (W32)→8: OP11 (W31)→9: OP16 (W31) →10: OP12 (W31)		
2. NLIP	<b>Module instances</b>	<b>Final layout</b>	<b>Total cost</b>	<b>Computation time</b>
	M1I2, M2I1, M3I2, M4I2	W1-W2-W3-W4	144.6€	3729.9s
	Process plan	1:OP1 (W22) →2: OP3 (W22)→ 3: OP6 (W14)→ 4: OP8 (W14)→ 5: OP7 (W21)→6: OP10 (W21)→ 7: OP14 (W32)→8: OP11 (W31)→ 9: OP16 (W31)→ 10: OP12 (W31)		

In order to test other cases in which machine displacement costs per time are not much higher than the other processing costs (i.e., material handling, operations, machine configuration change), a comparison with other lower displacement costs per time was carried out as presented in Table 12. Example 1 corresponds to the 40 % of the time multiplied by the cost of displacing machines stated in the first example (presented in Table 11), the second example corresponds to 10%, while the last one to 5%.

The results evidence the impact of machine displacement costs on the layout configuration. In the two first examples (i.e., 40% and 10%) layout should keep the same, meaning that changing machines' positions do not considerably contribute to reducing costs of material handling. In

contrast, in the third example (5%), the layout was changed, indicating that the new machines' positions contributed to reduce material handling costs. When displacing machines' costs are considerably higher than costs of reconfiguring machines or transporting material, the layout tend to keep its current configuration. However, when machine displacing costs are close to costs of handling material or reconfiguring machines, there is a good trade-off between the layout reconfiguration and the material handling.

Table 12. Results obtained by Exhaustive search-CPLEX on solving the NLIP problem, with displacement costs equal to 1) 40%, 2) 10% and 3) 5% of the initial displacement cost.

Exhaustive search-CPLEX				
<b>1. NLIP (40%)</b>	<b>Module instances</b>	<b>Final layout</b>	<b>Total cost</b>	<b>Computation time</b>
	M1I2, M2I1, M3I2, M4I2	W1-W2-W3-W4	<b>144,6 €</b>	<b>5194.1s</b>
	Process plan	1:OP1 (W22) →2: OP3 (W22)→ 3: OP6 (W14)→ 4: OP8 (W14)→5: OP7 (W21)→6: OP10 (W21)→ 7: OP14 (W32)→8: OP16 (W31)→9: OP12 (W31) →10: OP11 (W31)		
<b>2. NLIP (10%)</b>	<b>Module instances</b>	<b>Final layout</b>	<b>Total cost</b>	<b>Computation time</b>
	M1I2, M2I1, M3I2, M4I2	W1-W2-W3-W4	<b>144,6 €</b>	<b>5843.6s</b>
	Process plan	1:OP1 (W22) →2: OP3 (W22)→ 3: OP6 (W14)→ 4: OP8 (W14)→5: OP7 (W21)→6: OP10 (W21)→ 7: OP14 (W32)→8: OP16 (W31)→9: OP12 (W31) →10: OP11 (W31)		
<b>3. NLIP (5%)</b>	<b>Module instances</b>	<b>Final layout</b>	<b>Total cost</b>	<b>Computation time</b>
	M1I2, M2I2, M3I2, M4I3	W2-W1-W3-W4	<b>144,4 €</b>	<b>4914.8s</b>
	Process plan	1:OP1 (W22) →2: OP3 (W22)→ 3: OP5 (W11)→ 4: OP7 (W12)→ 5: OP10 (W14)→6: OP15 (W32)→ 7: OP13 (W32)→8: OP12 (W31)→ 9: OP16 (W31)→ 10: OP11 (W31)		

### 4.3. Evaluation of the initial machines' configurations changeovers

Some unpredictable factors can affect the search for the optimal solution in these mathematical problems, such as the initial machines' configurations and the functions selected by the customer. This section investigates how machines' initial configurations affect the optimization's results for different sets of functions.

In this example, each machine has a different number of configurations ( $W1=4$ ,  $W2=3$ ,  $W3=2$ , and  $W4=5$ ). If all machines' configurations are combined there are 120 possible initial configurations (i.e.,  $4 \times 3 \times 2 \times 5 = 120$ ). The initial machines' configurations were changed at a time while all other parameters were kept the same. All these 120 initial configurations were tested in the following layout configuration ( $W1:1$ ,  $W2:2$ ,  $W3:3$ , and  $W4:4$ ). Since the layout did not change, all tests were conducted using the ILP model. Table 13 presents a list fragment of the 120 initial configurations according to the machine-configurations available that were tested in this work.

Table 13. Fragment of the list of all possible initial machines' configurations.

Machine Configuration	W1				W2			W3		W4				
	C1	C2	C3	C4	C1	C2	C3	C1	C2	C1	C2	C3	C4	C5
Run 1	0	0	1	0	1	0	0	0	1	0	0	0	1	0
Run 2	0	0	1	0	1	0	0	0	1	0	1	0	0	0
Run 3	0	0	1	0	1	0	0	0	1	1	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Run 118	0	1	0	0	0	0	1	1	0	1	0	0	0	0
Run 119	0	1	0	0	0	0	1	1	0	0	0	0	0	1
Run 120	0	1	0	0	0	0	1	1	0	0	0	1	0	0

Besides the initial example presented in previous sections, in which functions 2, 6, 7, and 10 (group 1) were selected, two other groups were considered (Table 14) in order to understand how the initial machine's configuration impacts the total cost for different sets of required functions. In group 2, only one function changed compared to group 1, while in group 3, all required functions were different from group 1.

Table 14 summarizes the number of product candidates for each group of functions as well as the frequency these product variants appeared as an optimal solution within the 120 runs for each group. The product variants that most frequently appeared as the optimal solution (highlighted in grey) in each group presented the cheaper raw material costs and the lower number of operations of the corresponding group.

The results in Table 14 suggest that product variants requiring the same number of operations than the product variant that appeared most frequently, but about 6% more expensive than the latter, it will appear as optimal solution, but in a lower frequency (see product variants 1A and 1B). When the raw material cost of two product variants is the same, the product variant requiring one more operation will appear less frequently (see product variants 3A and 3B). When the other product variants require at least one more operation than that appearing most frequently and are



at least 3.5% more expensive than the reference product variant, they will never appear as an optimal solution (see product variants 1C and 1A; 2B,2C and 2A; 3C-3F and 3A).

In summary, these results show that different product variants tend to be the optimal solution depending on the initial machines' configuration, highlighting the benefits of concurrently optimizing the product configuration with the process planning in an RMS. The results suggest that product variants composed of module instances presenting cheaper raw material costs and requiring fewer operations tend to appear more frequently in an optimal solution considering the investigated layout configuration. Some product variants never appeared as an optimal solution because they present higher costs for raw material or require a higher number of operations than the product variant most frequently selected.

It is important to highlight that in this study only the total cost was considered. If other responses of interest are considered, such as customer satisfaction index, completion time, etc., some product variants that were not selected as optimal solutions in terms of cost in the examples presented could be selected as optimal solutions. Besides that, this study assumed that all machines and module instances were available; however, if the availability of machines and module instances are considered, the optimal solutions may change.

Table 14. Product variant candidates for each group of selected functions and the frequency they appeared as an optimal solution within the 120 runs.

Group	ID	Product variants	Frequency (%)	N° of required operations	Raw material cost (€)	Gap from the minimal cost (€)
<b>1 - functions 2, 6, 7, 10</b>	1A	M12, M22, M32, M43	75	10	52,6	0,00
	1B	<u>M12, M21, M32, M42</u>	25	10	55,76	6,01
	1C	M12, M21, M32, M43	0	11	55,26	5,06
<b>2 - functions 2, 6, 7, 9</b>	2A	M12, M22, M32, M41	100	9	50,8	0,00
	2B	M12, M21, M32, M41	0	10	53,46	5,24
	2C	<u>M12, M21, M32, M42</u>	0	10	55,76	9,76
<b>3 - functions 4, 5, 8, 9</b>	3A	M12, M21, M31, M41	87.5	10	42,66	0,00
	3B	M11, M21, M31, M41	12.5	11	42,66	0,00
	3C	M11, M21, M31, M42	0	11	44,96	5,39
	3D	M12, M21, M31, M42	0	10	44,96	5,39
	3E	M13, M21, M31, M41	0	11	44,16	3,52
	3F	M13, M21, M31, M42	0	11	46,46	8,91

Table 15 shows some statistics of the 120 runs for each group of functions. The results show how the number of product variants affects the computation time when the CPLEX-based approach is used. The higher the number of candidates, the higher the computation time. The computation time for group 3 was, on average, 391.41s. Although groups 1 and 2 have the same number of product variant candidates, the computation time for group 1 (170.62s) was slightly

higher than 2 (132.17s). This difference can be due to the higher number of operations required by product variants of group 1.

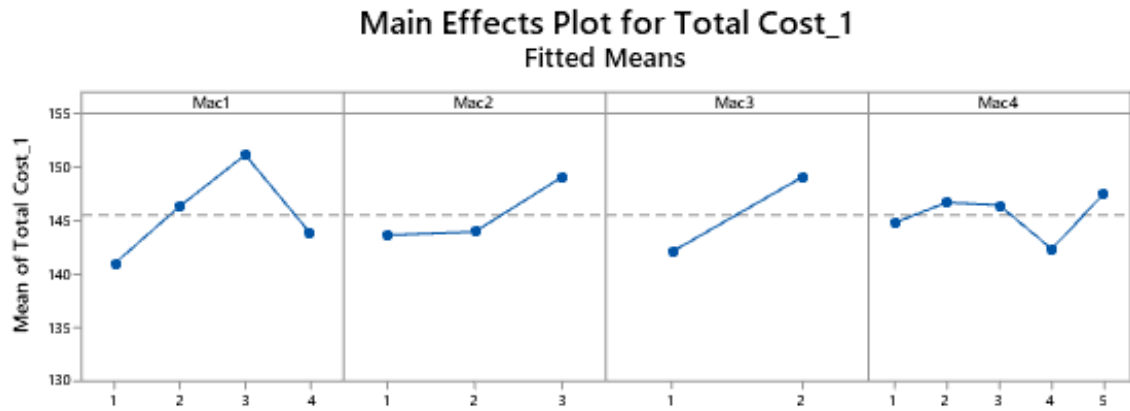
Table 15. Summary of the results obtained by each group of functions considering different initial machines' configurations.

<b>Group</b>	<b>1</b>		<b>2</b>		<b>3</b>	
<b>Functions</b>	<b>2, 6, 7, 10</b>		<b>2, 6, 7, 9</b>		<b>4, 5, 8, 9</b>	
<b>Responses</b>	<b>Total Cost (€)</b>	<b>Comp. Time (s)</b>	<b>Total Cost (€)</b>	<b>Comp. Time (s)</b>	<b>Total Cost (€)</b>	<b>Comp. Time (s)</b>
<b>Mean</b>	145.51	170.62	140.21	132.17	141	391.41
<b>StDev</b>	7.49	25.45	8.07	21.37	8.67	62.70
<b>Minimum</b>	126.31	134.8	122.49	107.13	120.73	289.78
<b>Median</b>	144.58	186.18	139.95	125.31	141.62	396.17
<b>Maximum</b>	164.52	228.13	159.49	159.47	159.28	490.56
<b>Gap</b>	30.25%	69.23%	30.21%	47.08%	31.94%	69.28%

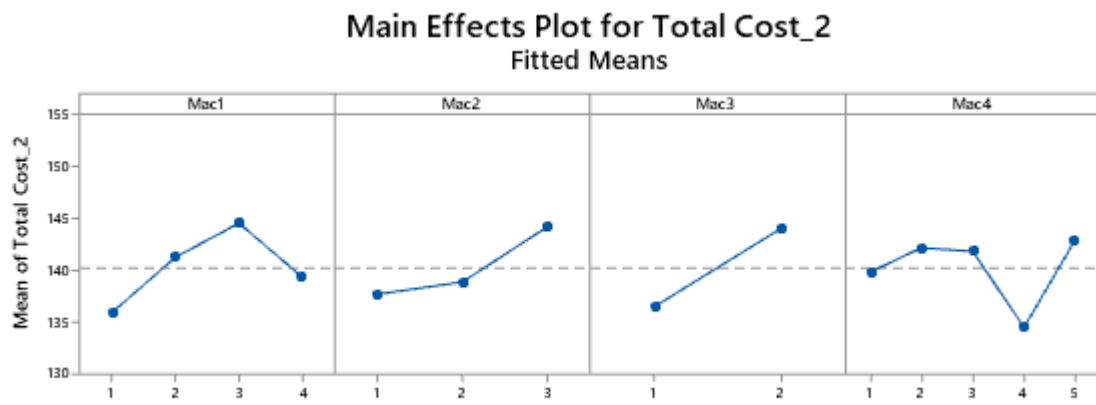
The results suggest that for all groups of functions considered, the total cost can vary up to about 30% for the layout configuration considered in this example, depending on the initial machines' configurations. The average cost for all groups was close to 140-145€. The minimal costs obtained by groups 1, 2, and 3 were 126.31€, 122.49€, and 120.73€. These minimal costs correspond to the most frequently selected product variants, according to data from Table 14.

Groups 1 and 2 attained the minimum cost when the machines started with the following configurations: W1C1, W2C2, W3C1, and W4C4. In contrast, group 3 attained the minimum cost when the initial machines' configurations were W1C2, W2C1, W3C2, and W4C2, an initial configuration utterly different from the other two groups. The similar behaviour between groups 1 and 2 is evidenced by the main effects plots of Figure 20a,b. They show that for groups 1 and 2 minimal total costs are obtained when machines W1, W2, W3, W4, respectively, start with the configurations C1, C1, C1, and C4. In contrast, for group 3, the main effects plot (Figure 20c) suggests that minimal costs are obtained when the initial configuration of machines W1, W2, W3, W4, are respectively, C4, C1, C2, C2. These results indicate a relationship between the type of functions required and the preferable initial machines' configurations.

a)



b)



c)

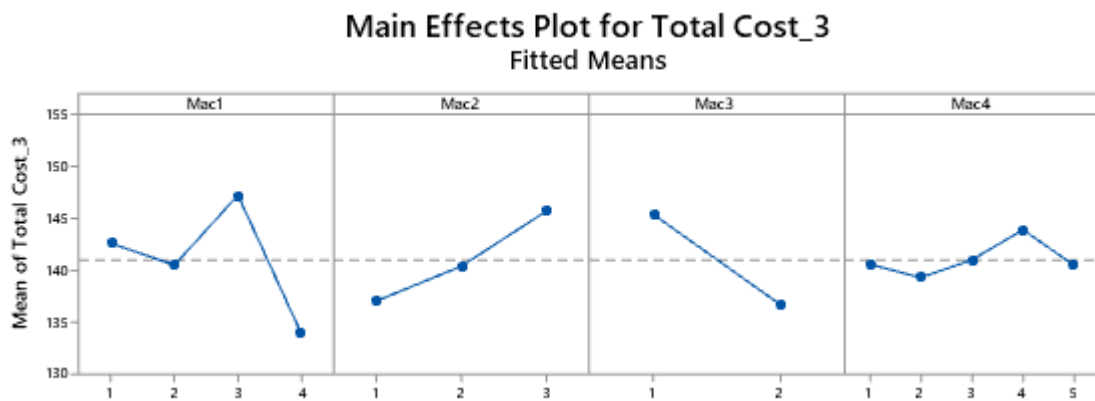


Figure 20. Main effects plot for the total cost of a) Group 1 b) Group 2, and c) Group 3.

#### 4.4. Resolution approaches validation

CPLEX-based approaches were useful to validate the mathematical model proposed and can be applied for small-sized examples. Nevertheless, for mid-large examples, CPLEX-based approaches, especially for solving nonlinear problems, become more time-consuming and then impracticable. This section aims to validate the two GA based- approaches detailed in Chapter 3 by comparing their respective results to the one obtained with CPLEX based approaches.

##### 4.4.1. GA parameters tuning

Although GA has proven its ability to solve a wide range of optimization problems, this does not guarantee that it has a good performance for solving all problems, including this one. As with all metaheuristics, a critical step of applying GA is its parameter tuning, which directly affects its performance.

In this work, the design of experiments technique was used to tune GA parameters. A full 2-level factorial design was planned for 10 replicates according to the parameters presented in Table 16, which resulted in 160 runs ( $10 \times 2^4 = 160$ ) for each approach. Besides the parameters presented in Table 16, there is the mutation rate ( $m$ ) that was changed during the experiments for ESA-GA. It was assumed that  $m = c - 1$ , meaning that when the crossover rate was 0.9, the mutation rate was 0.1, for example.

Table 16. Evaluated GA parameters and their tested levels for ESA-GA and mGA methods.

Parameters	Algorithm	Notation	Levels	
			-1	1
Crossover rate	ESA-GA	$c$	0.7	0.9
Ratio good ind. / random ind.	mGA	$r$	0.2	0.5
Tournament size	Both	$t$	3	7
Population size	Both	$p$	100	200
Number of generations	Both	$n$	50	100

Since mGA has an adaptive probability of crossover and mutation, these parameters were not considered as control factors for this approach. Instead, the ratio  $r$  of good individuals per random individuals were evaluated. It was considered as minimum and maximum equal to 0.2 and 0.5, respectively. This ratio corresponds to the fraction of good individuals in relation to the

total number of random individuals. For example, if  $r = 0.5$  and the total number of random individuals is 100, it means that the number of good individuals is equal to 50.

In both approaches, the levels of  $t$  varied from 3 to 7, population size  $p$  from 100 to 200 and number of generations  $n$  from 50 to 100. These levels were chosen for both algorithms because they have shown good performance in preliminary studies for both methods.

Both algorithms, mGA and ESA-GA, were tested separately only for the NLIP model (changeable layout) because preliminary tests evidenced that the same GA parameters were appropriate for solving both mathematical problems.

The computation time and best fitness of each run were archived and used to model the ‘best fitness function’ ( $f_{bf}$ ) and ‘computation time function’ ( $f_{ct}$ ), according to the parameters presented in Table 16. No lack-of-fit was found, indicating that the adjusted models correctly specified the relationship between predictors and the response. All functions as well as their adjusted  $R^2$  are presented in Table 17.

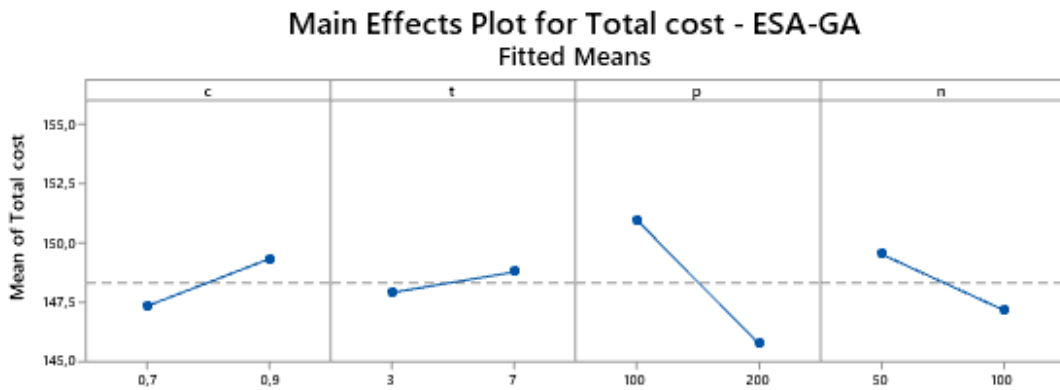
Table 17. Functions and their adjusted  $R^2$ .

Function	ESA-GA	mGA
$f_{bf}$	$0,4913 + 0,0595 c + 0,00002 t - 0,000091 p - 0,000320 n + 0,00580 c * t - 0,000370 c * p - 0,000053 t * n + 0,000003 p * n$	$0.2943 + 0.0773 r + 0.05575 t + 0.001199 p + 0.002710 n - 0.000820 r * p - 0.00144 r * n - 0.000292 t * p - 0.000600 t * n - 0.000016 p * n + 0.000012 r * p * n + 0.000003 t * p * n$
$R^2_{adj}$	55.50%	52.06%
$f_{ct}$	$12,47 - 0,0614 p - 0,111 n + 0,006964 p * n$	$3,18 - 0,0034 p - 0,023 n + 0,007752 p * n$
$R^2_{adj}$	89.02%	94.58%

Figure 21 presents the main effects plots for the best fitness function obtained by each method. The plots show that the higher levels of the number of generations ( $n$ ) implied in minimal costs in both methods. The low level of tournament size ( $t$ ) and population size ( $p$ ) also implied lower costs for both methods, but these effects have a higher importance in the mGA method. For the ESA-GA, values close to the lower level of crossover rate ( $c$ ) led to lower costs. In the mGA, when initial populations are composed by a rate of good individuals corresponding to the 50% of random individuals it attains lower best costs.

The mean of costs obtained by ESA-GA (148.32) is lower than the mean of costs obtained by mGA (151.18), represented by the grey pointed-line, indicating the overall mean of costs obtained by ESA-GA was closer to the global minimum (144.58€) than mGA.

a)



b)

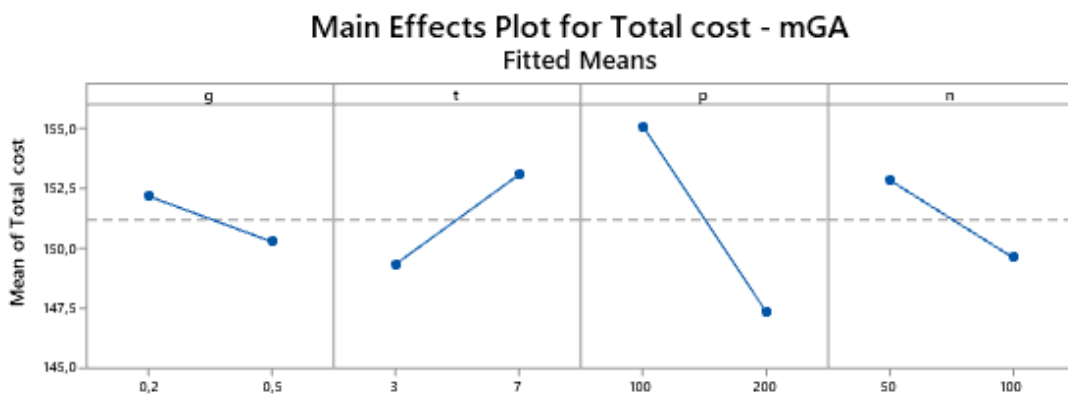
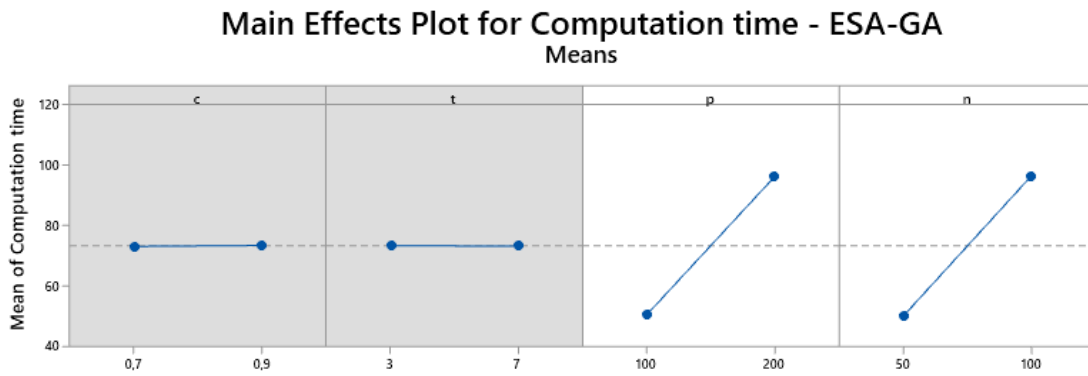


Figure 21. Main effects plot for best fitness for method a) ESA-GA and b) mGA.

The only terms statistically significant on the computation time function of both methods (Table 17) are population size (p) and number of generations (n). The factorial plots for the main effects for computation time confirm this (Figure 22). The value of good individuals (g), the crossover rate (c), and the tournament size (t) are parallel to the x-axis, indicating that the computation time mean is the same for all factors' levels, and then, there is no main effect present for these factors.

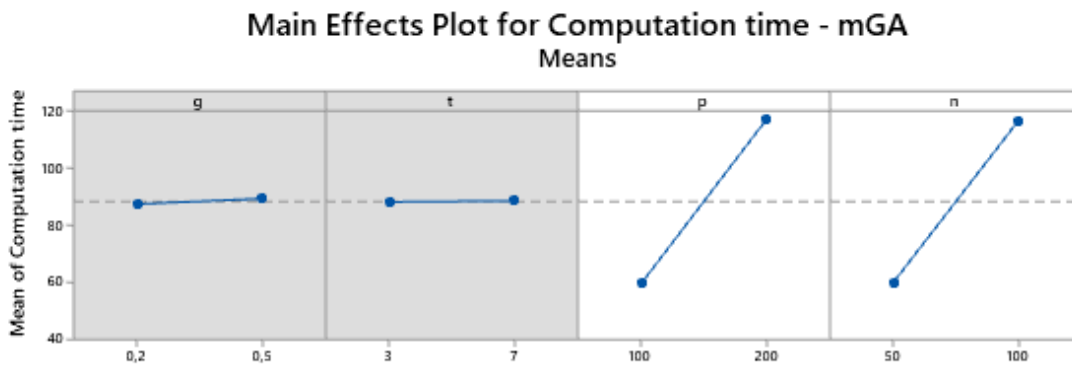
The lower computation times were obtained for lower levels of population size and number of generation, contrasting to these effects behaviour for costs since lower costs were obtained for higher levels of p and n. The mean effects of number of generations appear to be a higher magnitude in computation time than in costs.

a)



A gray background represents a term not in the model.

b)



A gray background represents a term not in the model.

Figure 22. Main effects plot for computation time for method a) ESA-GA and b) mGA.

After properly modelling the functions from Table 17 and evaluating the main effects of each function individually for each method, the optimal GA parameters were obtained through the solution of the optimization problem presented in Equations (58).

$$\begin{aligned}
 & \text{Min } f_{global} = f_{bf} + f_{ct} \\
 & \text{s.t.:} \\
 & \quad 100 \leq p \leq 200 \\
 & \quad 50 \leq n \leq 100 \\
 & \quad 3 \leq t \leq 7 \\
 & \quad 0.7 \leq c \leq 0.9 \quad \text{or} \quad 0.2 \leq r \leq 0.5 \\
 & \quad \text{ESA-GA} \quad \quad \quad \text{mGA}
 \end{aligned} \tag{58}$$

The optimization problem was solved through the desirability method of Minitab19®. The optimization results are presented in Table 18. The values from Table 18 were used to tune de ESA-GA and mGA for optimizing the ILP and NLIP for the small example, previously solved with the CPLEX-based approaches in Section 4.2.

Table 18. Optimal GA parameter values according to the method and problem type.

Method	ESA-GA	mGA
Crossover rate	0.7	-
Mutation rate	0.3	-
Ratio good/random ind.	-	0.5
Population size	200	100
Number of generations	50	50
Tournament size	3	3
Composite desirability (D)	0.8718	0.9055

#### 4.4.2. Applying optimal GA parameters

After tuning GA parameters, each algorithm (mGA and ESA-GA) ran 100 independent times, for solving each problem (i.e., ILP and NLIP), with random seeds for ESA-GA, while mGA was initialized with random and good individuals at a time. The example used here is the same used to validate the mathematical model in Section 4.2, as detailed in Table 10.

##### 4.4.2.1. Solving the ILP problem (Proposition 1)

Table 19 summarizes the results found for the first mathematical problem (Proposition 1), in which product configuration is concurrently optimized with the process planning. As expected, the optimal solution of both methods selected the same product variant leading to the global minimum cost of 144.6€, as the one obtained by the CPLEX-based approach.

We can be 96.3% confident that the optimal results found by ESA-GA and mGA will be between 144.576 and 151.856, and 144.576 and 164.959, respectively. Meaning that the optimal attained by ESA-GA and mGA can be up to 5.0% and 14.1% higher than the optimum global cost (144.6€). This gap is also evidenced by the standard deviation ( $\sigma$ ) since the  $\sigma$  of mGA was more than 3 times higher than the standard deviation of ESA-GA, indicating that ESA-GA is more reliable than mGA.

Despite that, both methods attained the global optimum in at least 50% of the runs since their median was 144.6€. From that it is possible to infer that the higher standard deviation of mGA is mainly due to outliers values, which can be a consequence of the situations in which the algorithm was trapped in local optima. Since all decisions are integrated into mGA, including the product



configuration part, GA has a higher search space to explore. As a consequence, the possibility of being trapped into bad solutions is higher.

Table 19. Results obtained for the *ILP* problem.

	Method	ESA-GA			mGA		
Statistical parameters for the 100 runs data		Average	Median	Standard deviation	Average	Median	Standard deviation
	Cost (€)	144.9	144.6	1.01	146.6	144.6	3.75
	Time (s)	70.1	62.2	21.1	40.3	40.9	2.58
		Tolerance interval for 95% of population		Confidence level (%)	Tolerance interval for 95% of population		Confidence level (%)
	Cost (€)	(144.576, 151.856)		96.3	(144.576, 164.959)		96.3
	Time (s)	(55.316, 142.622)		96.3	(34.071, 57.577)		96.3
Optimal results		Cost (€)	Time(s)		Cost (€)	Time(s)	
		144.6	55.3		144.6	34.1	
Module instances		M12, M21, M32, M42			M12, M21, M32, M42		
Process plan		1:OP1(W2C2) → 2: OP3( W2C2) → 3: OP6( W1C4) → 4: OP8( W1C4) → 5: OP7( W2C1) → 6: OP10( W2C1) → 7: OP14( W3C2) → 8: OP11( W3C1) → 9: OP12( W3C1) →10: OP16( W3C1)			1:OP1(W2C2) → 2: OP3( W2C2) → 3: OP6( W1C4) → 4: OP8( W1C4) → 5: OP7( W2C1) → 6: OP10( W2C1) → 7: OP14( W3C2) → 8: OP16( W3C1) → 9: OP11( W3C1) →10: OP12( W3C1)		

Regarding the computation time, mGA is almost 2 times faster than ESA-GA, keeping a  $\sigma$  close to 2.6s, while the standard deviation for computation time in ESA-GA is about 21.1s. As previously stated, all decisions (i.e., product configuration and process planning) are simultaneously made by mGA; therefore the computation time for this method does not depend on the number of product variant candidates.

In contrast, in ESA-GA, the product configuration decision is previously made through the ESA and each process plan is optimized for each product variant individually. As a consequence, in the ESA-GA, the higher the number of product variants, the higher the computation time. In the example used here, three product variants are capable of satisfying the set of FRs selected, which partially explain the higher average computation time for ESA-GA. Besides that, mGA was tuned with  $p=100$ , while ESA-GA was tuned with  $p=200$ . As previously shown, higher levels of population size lead to higher means of computation time. Therefore, the higher computation time of ESA-GA can also be explained by its higher population size.

The average computation times of ESA-GA and mGA are lower than that of the CPLEX-based approach (122s), but the three approaches' average computation times are relatively low.

Since this consists of an ILP problem, CPLEX is effective on solving it. Considering the relatively short computation time obtained for a small example, there is no interest in using the evolutionary approaches for this small case of ILP because CPLEX-based approach presents a relatively good performance and ensures that the global optimum will always be obtained.

Nevertheless, this comparison is useful because it indicates the ability of the evolutionary methods to solve the ILP problem presented in Proposition 1 since both methods were capable of obtaining the global optimum. The CPLEX-based approach can work well for small-sized problems, but it may not be able to optimize problems with larger instances, and then, impracticable for solving real-life problems. Both evolutionary approaches, in contrast, can solve problems with higher instances. Therefore, both approaches will be tested in the next chapter through two illustrative examples based on real products.

Finally, regarding the process plans obtained by both methods, they were very similar. Only operations with no precedence relationship being executed in the same configuration of the same machine (W3C1) varied their sequence in the process. Despite this difference in operations sequence, the total cost kept the same because there is no cost of changing the operation when machine-configuration is not changed.

#### **4.4.2.2. Solving the NLIP problem (Proposition 2)**

Table 20 presents the results obtained by ESA-GA and mGA for the model presented in Proposition 2, in which product configuration is concurrently optimized with the process planning and layout configuration (NLIP model).

ESA-CPLEX spent 3729.9s to get a solution while ESA-GA and mGA respectively spent, on average, 64.7s and 40.4s. The mGA was about 1.5 times faster than ESA-GA and more than 90 times faster than the CPLEX-based approach. While both GA-based approaches kept their computation time close to the computation time required for solving the ILP problem, the computation time required by the ESA-CPLEX approach increased almost 30 times.

As previously detailed the ESA-CPLEX approach for solving NLIP problem, requires an additional step (Step 3). ESA-CPLEX spends much more time solving the NLIP problem than the ILP one because each product variant's process planning is optimized individually for each layout configuration. In the problem addressed here, the 4 available machines correspond to 24 possible layouts. Hence, this approach iterates over the 24 layout configurations, and in each one, the process planning of each of the three product variant candidates is optimized individually.

This explains why the computation time for ESA-CPLEX was much higher in this problem, than in the previous one.

Table 20. Results obtained for the *NLIP* problem.

	Method	ESA-GA			mGA		
Statistical parameters for the 100 runs data		Average	Median	Standard deviation	Average	Median	Standard deviation
	Cost (€)	145.6	144.6	1.75	150.9	147.6	7.07
	Calc. Time (s)	64.7	59.5	16.34	40.4	39.2	3.36
		Tolerance interval		Confidence level (%)	Tolerance interval		Confidence level (%)
	Cost (€)	(144.576, 158.540)		96.3	(144.576, 168.367)		96.3
	Calc. Time (s)	(55.636, 160.401)		96.3	(34.263, 57.709)		96.3
Optimal results	Cost (€)	Calc. Time(s)		Cost (€)	Calc. Time(s)		
		144.6	55.6	144.6	34.3		
Module instances	M12, M21, M32, M42			M12, M21, M32, M42			
Process plan	1:OP1(W2C2) → 2: OP3( W2C2) → 3: OP6(W1C4) → 4: OP8( W1C4) → 5: OP7(W2C1) → 6:OP10( W2C1)→ 7: OP14(W3C2) →8: OP11(W3C1)→ 9: OP12( W3C1) →10: OP16( W3C1)			1:OP1(W2C2) → 2: OP3( W2C2) → 3: OP6( W1C4) → 4: OP8( W1C4) → 5: OP7( W2C1) → 6: OP10( W2C1) → 7: OP14( W3C2) → 8: OP16( W3C1) → 9: OP11( W3C1) →10: OP12( W3C1)			
Final layout	W1:L1; M2:L2; W3:L3; W4:L4			W1:L1; M2:L2; W3:L3; W4:L4			

If ESA-CPLEX spent more than one hour to solve a small problem with 4 machines, it would require even more time to solve a problem with 5 or more machines ( $5! = 120$  layouts;  $6! = 720$  layouts, and so on), being very time-consuming and impracticable in real-life problems. This reinforces the need to use non-deterministic approaches, such as the GA-based approaches proposed here.

The optimal cost found in this problem was the same as the ILP. In this problem, the optimal solution suggests that the layout configuration should not be changed; therefore, all machines kept in their current positions, and the machines' displacement cost was null. Once again, the process planning obtained by both approaches was similar, presenting differences only in the operations' sequence of operations being processed in machine W3 with configuration C1.

Although both GA-based methods attained the global optimum cost found in ESA-CPLEX in much less time, ESA-GA appeared to be more reliable than mGA since the standard deviation

of the cost results was only 3.12€, and the median was equal to 144.6€, implying that at least 50% of the results attained the global optimum. In contrast, mGA's median was 147.6€, implying that less than 50% of the runs attained the global optimum. In addition to that, the standard deviation of mGA was also more than two times higher than the  $\sigma$  of ESA-GA.

Regarding the future results, we can be 96.3% confident that at least 95% of future ESA-GA results will be between 144.576€ and 158.540€, and for mGA, they will be between 144.576€ and 164.040€ when functions 2, 6, 7, and 10 are required. Therefore, ESA-GA and GA can find solutions up to 9.7% and 13.5% more expensive than the optimal one in much less time than the CPLEX-based approach.

It is up to the decision-maker to decide if this difference of costs is significant or not regarding the additional time required to optimize this problem using the ESA-CPLEX approach. If not significant, then GA-based approaches appear to be preferable to solve this problem. ESA-GA appeared to be more reliable than mGA; therefore, for small problems or in case that there is not a wide number of product variant candidates, the ESA-GA can be preferable. However, in case that the number of product variant candidates is very large and computation time is a constraint, mGA can be preferable than the ESA-GA. In any case, the choice will depend on the number of product variants and the constraints imposed on the computation time.

In summary, the analyses presented here indicate that both GA-based approaches have a good ability to solve higher size problems in less computation time. The next chapter will present two real-product based examples to illustrate how these approaches will perform for solving mid/large problems.

#### **4.5. Final considerations from this chapter**

This chapter validated both mathematical models proposed in Chapter 3 by using deterministic approaches based on exhaustive search and CPLEX solver. The results confirmed that the mathematical models correctly performed under the initial modelling assumptions while satisfying all constraints.

Some analyses were carried out to investigate the impact of layout reconfiguration change costs, which are directly dependent on the costs per time of displacing machines. The results suggest that the layout only should be reconfigured when repositioning machines can significantly reduce material handling costs, compensating the additional costs of displacing machines. This trade-off generally occurs when displacing machines' costs are not substantially higher than the costs of handling material and changing machines' configurations.

Regarding the current manufacturing industries, displacing machines may not be viable due to their weight and size. Nevertheless, recently with the development of cobotic technologies and automatic guided vehicles (AGV), it becomes possible to easily displace some machines (or robots) to perform operations in different layout positions.

Further, considering micro-factories endowed with smart surfaces, displacing machines is as easy as transferring materials between them. In these cases, concurrently minimizing the costs of product configuration with process planning and layout reconfiguration seems to be the best strategy. Nevertheless, if machines cannot be easily displaced, the ILP model should be used to optimize the product configuration concurrently to the process planning.

The influence of initial machines' configurations was also investigated in this chapter. An analysis was carried out to understand how the initial machines' configurations affected the total cost for different groups of functions in a given layout. Different groups of functions presented a gap close to 30% within the minimum and maximum costs considering all possible initial machines' configurations. This shows how RMS's current configuration can impact the manufacturing costs and the importance of considering it.

Different product variants appeared as optimal solutions depending on the initial machines' configurations, confirming the initial RMS configuration's influence on the product configuration selection and the benefits of concurrently optimizing the product configuration and the process planning in an RMS. Some product variants never appeared as an optimal solution for a given layout; however, it does not mean that they cannot be an optimal solution in other layout configurations.

After evaluating the impact of the initial layout and machines' configurations on the total manufacturing costs, the GA parameters of the two GA-based approaches proposed in Chapter 3 were tuned for solving the small-sized example introduced in this chapter. Both methods ran 100 independent times, and their results were compared with those obtained by the CPLEX-based approach. Both methods have proven their ability to solve the ILP and NLIP models in less computation time than the CPLEX-based approaches. In Chapter 5, both GA-based approaches will be used to solve two mid-size illustrative examples based on real products.

## CHAPTER 5 - CASE STUDIES

This chapter aims to apply the GA-based methods for solving mid-large examples that illustrate real product applications. Two types of modular products were selected to illustrate the method: a smartphone and a sneaker. The following sections will present more details about each example. Only NLIP problem is addressed in the illustrative examples. The data used in these case studies is estimated. All tests were conducted on a server powered by 2 Intel Xeon Platinum 8160 X7542 24-core, 48 thread CPUs (2.1 GHz) and 1To of RAM.

### 5.1. Case 1: Modular smartphone

Smartphone varieties and functionalities are increasingly growing. With a large set of choices, customers face the challenge of choosing which smartphone could better meet their needs. Therefore, manufacturers must perceive these needs and translate them into a product capable of satisfying all specific customer demands.

Further, smartphone features' descriptions can sometimes be incomprehensible by non-experts or customers new to a specific product category. The approach presented here aims to show how it can be feasible and relatively easy for the customers to find a product capable of meeting their requirements when asked about what they want in their product or how they want to use it instead of directly choosing among features proposition.

The smartphone used to illustrate this example contains 7 modules (total of 20 module instances), which are described in Table 21, 26 available product functions, 7 machines (with a total of 23 machine-configurations) and 32 operations (see Appendix B for more information about this example). Figure 23 illustrates a modular smartphone variant, which is composed by one base module that can have three colour variants and other modules with different functionalities (Table 21).

Table 21. Modules and module instances of the modular smartphone.

Modules		M1	M2	M3	M4	M5	M6	M7
		Base module	Camera	Connectivity	Internal memory	Capacitive touchscreen	Battery	System on a Chip (SoC)
Module instances	I1	Black	16 MP	Wi-Fi	6 GB RAM	LCD	5000 mAh	Quad-core 2,2 GHz +chipset 10nm
	I2	Grey	12 MP	Bluetooth	12 GB RAM	OLED	3400mAh	Hexa-core 2,2 GHz +chipset 10nm
	I3	White	6 MP	Both	-	-	-	Octa-core 2,73 GHz +chipset 7nm
	I4	-	None	None	-	-	-	-

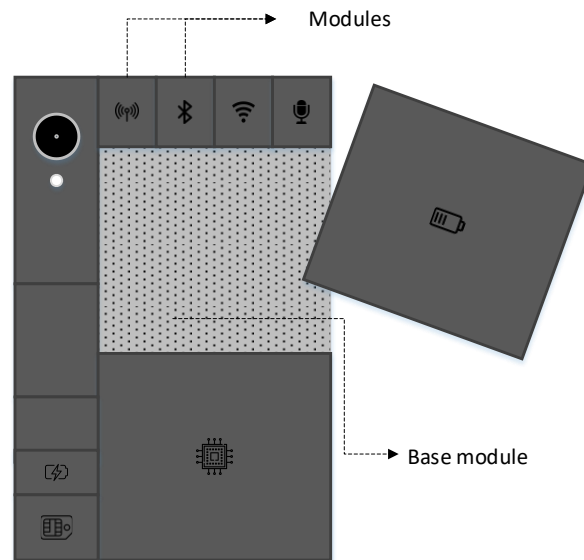


Figure 23. Modular smartphone.

The functional requirements (FRs) customer can select as well as the module instances capable of satisfying them are described in Table 22. The functions selected by the customer in this example were 4, 6, 11, 15, 18, 21, and 24.

Table 22. Module instances capable of satisfying each available functional requirement.

Function	Function description	Can be satisfied by module instance(s):
F1	Prefers a black smartphone	M11
F2	Prefers a grey smartphone	M12
F3	Prefers a white smartphone	M13
F4	Has no preferences concerning the smartphone colour	M11   M12   M13
F5	Frequently takes pictures with the smartphone. Needs high quality pictures.	M21
F6	Sometimes takes pictures with the smartphone. Does not need high picture quality	M21   M22
F7	Rarely takes pictures with the smartphone, does not worry about the quality. However, she/he thinks that it is important to have a camera in the smartphone.	M23
F8	Never uses smartphone camera, thus prefers not to have a camera in her/his smartphone.	M24
F9	Has a connected electronic device (e.g., smartwatch, etc.) that connects with the smartphone through Bluetooth. Needs internet access.	M33
F10	Has a connected electronic device (e.g., smartwatch, etc.) that connects with the smartphone through Bluetooth. Does not need internet access.	M32
F11	Does not have connected electronic devices neither transfer documents through the Bluetooth, but needs internet access.	M31
F12	Does not have connected electronic devices neither transfer documents with Bluetooth. Do not need internet access.	M34
F13	Usually, installs several applications on the smartphone.	M42
F14	Usually, saves many documents (especially pictures) on the smartphone.	M42
F15	Installs some applications and uses the smartphone to save some, but not many pictures.	M41   M42
F16	Does not need to install many applications either pictures.	M41
F17	Does not need good screen resolution because she/he just uses the smartphone to call or send some messages.	M51   M52
F18	Sometimes watches videos on internet but high resolution is not really important.	M51   M52
F19	Often watches movies, play games and likes to see pictures on the smartphone.	M52
F20	Rarely uses the smartphone during the day, just to call and send messages.	M61   M62
F21	Sometimes use the smartphone during the day to browse the internet.	M61
F22	Frequently calls, sends messages, takes pictures, plays games and browses on the internet.	M61
F23	Uses the smartphone to do the basic (i.e., to call and send messages)	M71   M72   M73
F24	Uses the smartphone to do the basic, but also likes to browse on the internet and watch some videos. Does not have many installed applications.	M71   M72
F25	Usually installs many applications on smartphone and uses them simultaneously.	M73
F26	Uses the smartphone like a small computer: installs many applications, plays games, and watches movies. Often opens many applications simultaneously.	M73



Table 23 summarizes the input parameters values considered in this illustrative example.

Table 23. Input parameters values.

Required product functions	Initial machines' configurations	Initial layout configuration
4, 6, 11, 15, 18, 21, 24	W1C4, W2C2, W3C2, W4C2, W5C5, W6C2, W7C1	W1: L1, W2: L2, W3: L3, W4: L4, W5: L5, W6: 6L, W7: L7

Similar to the small-sized example presented in Chapter 5, the Design of Experiments was used here to tune the GA parameters. A full 2-level factorial design was planned for 10 replicates according to the 4 parameters presented in Table 24 ( $10 \times 2^4 = 160$  runs). Mutation rate ( $m$ ) corresponded to  $1 - c$ .

Table 24. Evaluated GA parameters and their correspondent tested levels for smartphone's case study.

Parameters	Algorithm	Notation	Levels	
			-1	1
Crossover rate	ESA-GA	$c$	0.7	0.9
Tournament size	Both	$t$	3	7
Population size	ESA-GA	$p$	200	600
Population size	mGA	$p$	500	800
Number of generations	ESA-GA	$n$	200	300
Number of generations	mGA	$n$	250	400
Ration good/ random individuals	mGA	$r$	0.2	0.5

The best fitness and the computation time of each run, for each method were used for modelling the cost ( $f_{bf}$ ) and time ( $f_{ct}$ ) functions presented in Table 25 with the aid of Minitab19®.

Table 25. Functions modelled and their R<sup>2</sup> adjusted for mGA and ESA-GA.

Function	ESA-GA	mGA
$f_{bf}$	$0.2516 + 0.0802 c -$ $0.004312 t + 0.000179 p +$ $0.000302 n + 0.00484 c * t -$ $0.000245 c * p - 0.000384 c *$ $n + 0.000001 t * p -$ $0.000001 p * n + 0.000001 c * p *$ $n$	$0.3956 - 0.266 r - 0.02072 t -$ $0.000033 p - 0.000204 n + 0.0645 r *$ $t + 0.000248 r * p + 0.000750 r * n +$ $0.000016 t * p + 0.000076 t * n +$ $0.000000 p * n - 0.000073 r * t * p -$ $0.000220 r * t * n - 0.000001 r * p *$ $n - 0.000000 t * p * n + 0.000000 r *$ $t * p * n$
R <sup>2</sup> <sub>adj</sub>	62.34%	52.46%
$f_{ct}$	$-49944 + 52771 c + 7319 t +$ $82.3 p + 265.7 n - 7739 c * t -$ $83.3 c * p - 285 c * n - 13.95 t *$ $p - 40.3 t * n - 0.561 p * n +$ $14.4 c * t * p + 43.7 c * t * n +$ $0.631 c * p * n + 0.0993 t * p *$ $n - 0.1079 c * t * p * n$	$-3 - 577 r + 9.68 t - 0.460 p -$ $0.333 n + 0.551 r * p + 1.643 r * n +$ $0.005408 p * n$
R <sup>2</sup> <sub>adj</sub>	68.91%	93.65%

The Minitab's desirability method simultaneously optimized both functions to find the GA parameters capable of minimizing the overall manufacturing cost with minimal computation time. The optimal parameters suggested by the results obtained for each method are detailed in Table 26.

Table 26. Optimal GA parameter values.

Method	ESA-GA	mGA
<b>Crossover rate (c)</b>	0.7	-
<b>Mutation rate (m)</b>	0.3	-
<b>Population size (p)</b>	600	800
<b>Number of generations (n)</b>	300	250
<b>Ratio good/random ind. (r)</b>	-	0.5
<b>Tournament size (t)</b>	3	7
<b>Composite desirability (D)</b>	0.7144	0.9685

These values were used to tune parameters of each GA-based method, and then, both individually ran 100 independent times. The results obtained are summarized in Table 27.

The optimal solution detailed for each method corresponds to that with the minimal overall manufacturing cost among the 100 results. The optimal computation time represents the minimal one among the 100 runs and it does not necessarily correspond to the time spent by the optimal solution in terms of cost minimization.

Table 27. Results obtained for the modular smartphone.

	Method	ESA-GA			mGA		
Statistical parameters for the 100 runs data		Average	Median	Standard deviation	Average	Median	Standard deviation
	Cost (€)	602.5	603.2	6.897	626.6	625.9	15.107
	Comput. Time (s)	7956.9	7701.9	1588.1	925.0	927.4	161.7
		Tolerance interval for 95% of population		Confidence level (%)	Tolerance interval for 95% of population		Confidence level (%)
	Cost (€)	(587.111, 617.923)		95	(592.862, 660.355)		95
	Comput. Time (s)	(5353.825, 11454.573)		96.3	(664.035, 1351.434)		96.3
Optimal results	Cost (€)	Comput.. Time(s)		Cost (€)	Comput.. Time(s)		
	586.6	5982.4		594.4	664.0		
Module instances	M1I1, M2I2, M3I1, M4I2, M5I2, M6I1, M7I1			M1I1, M2I2, M3I1, M4I2, M5I2, M6I1, M7I1			
Process plan	1:OP10(W3C2) → 2: OP17( W3C2)→ 3: OP22( W2C2)→4: OP9( W2C3)→ 5: OP4( W2C3)→6: OP5( W2C2)→ 7: OP1( W1C1)→8: OP3( W1C3)→ 9: OP7( W2C3)→10: OP23( W4C2)→ 11: OP27(W4C2)→12:OP11(W4C1) → 13: <b>OP15(W4C1)</b> →14: OP24(W4C3)→ 15: OP26(W4C2)→16: OP28(W4C2)→ 17: OP18(W5C2)→18: OP20(W5C4)→ 19: OP21(W5C4)			1:OP17(W3C2)→2: <b>OP15(W3C2)</b> → 3:OP10(W3C2)→4:OP22(W2C2)→ 5:OP9(W2C3)→6:OP4(W2C3)→ 7:OP5(W2C2)→8:OP27(W4C2)→ 9:OP18(W5C2)→10:OP20(W5C4)→ 11:OP21(W5C4)→12:OP23(W4C2)→ 13:OP26(W4C2)→14:OP28(W4C2)→ 15:OP24(W4C3)→16:OP11(W4C1)→ 17:OP7(W2C3)→18:OP1(W1C1)→ 19:OP3(W1C3)			
Final layout	W1:L1; W2:L2; W3:L3; W4:L4; W5:L5; W6:L6; W7:L7			W1:L1; W2:L2; W3:L3; W4:L4; W5:L5; W6:L6; W7:L7			

The optimal solutions found for each method indicate that the best optimal product variant is that composed by the following module instances M11, M22, M31, M42, M52, M61, M71. This corresponds to a black (M11) smartphone, with a 12MP camera (M22), 12 GB of internal memory (M42) and Wi-Fi connectivity (M31). This smartphone is also equipped with a system on a chip Quad-core 2.2GHz+10nm chipset (M71), a 5000mAh battery (M61) and an OLED capacitive touchscreen (M52). These results confirmed the ability of both methods to find a product variant capable of satisfying all required functions.

Although operations sequence varied between both optimal process plans, most of the operations were performed using the same machine-configuration in both process plans, except for operation 15 (highlighted in bold). It means that the main reason for the difference of costs between both optimal solutions obtained by ESA-GA and mGA is due to the costs of machine's configuration changeovers and material handling. This emphasizes how costs of operations,

material handling, and machine configuration changes are interrelated, directly affecting the overall manufacturing cost and the operations sequence.

In both cases, the optimal solution suggests that the layout should keep its initial configuration. Moreover, all other 99 results obtained by each method also suggest that the layout should not be modified. This is expected since when the costs of displacing machines are considerably higher than that of changing machine’s configuration or handling material, it is preferable to keep machines in their current position.

ESA-GA attained an optimal cost of 586.6€, which is 1.3% lower than that obtained by mGA (594.4€). Globally, ESA-GA appears to outperform mGA in terms of cost minimization since it attained, on average, 602.5€ as optimal cost, which is about 10% lower than the average costs obtained by mGA (626.6€). The plot in Figure 24 shows that the population mean of results obtained by ESA-GA is within 601.149€ and 603.886€ and for mGA the cost mean is within 623.611€ and 629.606€ at 95% of confidence. The confidence intervals of mGA and ESA-GA distributions do not overlap, indicating that the difference of their means is statistically significant.

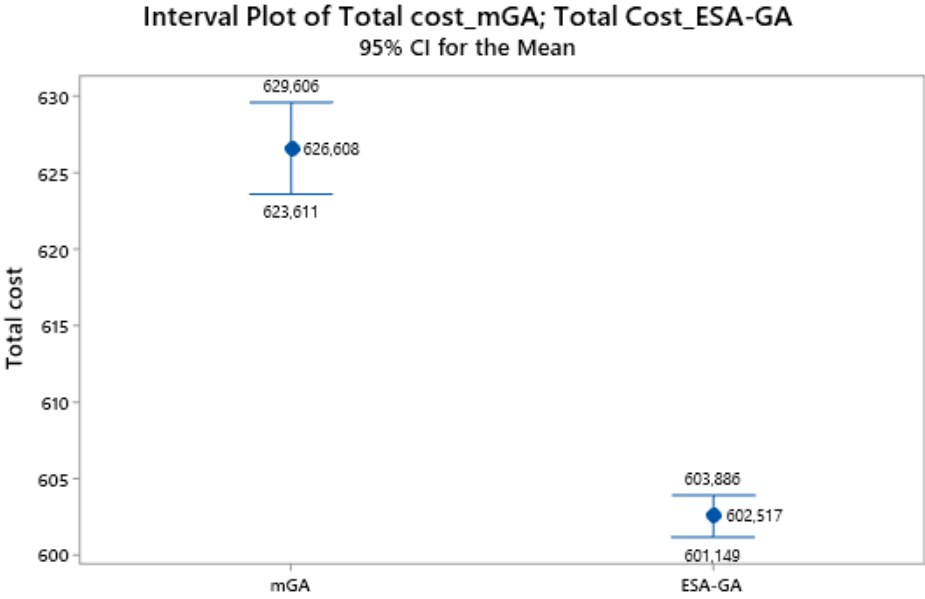


Figure 24 Interval plot of total costs obtained by mGA and ESA-GA in the smartphone case study.

A 2-sample t-test was carried out to investigate how significant is the difference of means between ESA-GA and mGA (Table 28). The results show that there is a significant difference between both methods at the 0.05 level of significance of  $|24.091|$ , and we can be 95% confident

that the true difference is between -27.375 and -20.807. Therefore, it is possible to affirm that ESA-GA outperformed mGA in terms of cost minimization.

For future runs, we can be 95% confident that at least 95% of results obtained by ESA-GA will be between 587.111€ and 617.923€ ( $\Delta=30.812\text{€}$ ), while for mGA these values will keep between 592.862€ and 660.355€ ( $\Delta=67.493\text{€}$ ).

Table 28. Anderson Darling and 2 Sample t-tests for the distributions of total cost obtained by ESA-GA and mGA for the smartphone case study.

Total cost	Index	Anderson Darling Normality Test
ESA-GA	1	p=0.283
mGA	2	p=0.849
2 Sample t-Test		
P-value (p)	Point estimate	CI : 95%
p < 0.05	$\bar{x}_1 - \bar{x}_2 = -24.091$	$\bar{x}_1 - \bar{x}_2 = (-27.375, -20.807)$

As introduced in the previous chapter, mGA makes all decisions into the chromosome evolution, while for ESA-GA, the decision made by GA (step 2) is restricted to the process planning and layout configuration for each product variant found by ESA in step 1. This considerably reduces the solution space for step 2 of ESA-GA and may explain why ESA-GA presented lower cost gaps, with lower standard deviation, and attained a minimal cost lower than mGA.

Even though ESA-GA outperforms mGA in terms of cost minimization, the latter appears to be much more performant regarding the computation time since it is about 8 times faster than ESA-GA. 50% of the runs for mGA spent up to 927.4s (15.46 minutes), while for ESA-GA the half of runs spent up to 7701.9s (2.14 hours).

Although both computation times presented large standard deviations, that of ESA-GA was almost 10 times higher than the variation of mGA. This variation is evidenced by the difference between the minimum and maximum computation times. While for ESA-GA there is a gap of almost 2 hours between the lower and higher computation time, for mGA this difference corresponds only to approximately 11.5 minutes.

As previously said, there are 24 product variant candidates for the example used in this case study. ESA-GA is very sensitive to the number of product variants capable of satisfying a set of customer requirements; the higher the number of product variants, the higher the computation time, explaining why the computation time required by ESA-GA was much higher

than that of mGA. In the example considered here, ESA-GA took up to 11454.573s (~3.2 hours) to obtain the optimal result, while mGA took up to 1351.434s (~22.5 minutes). We can be 96.3% confident that at least 95% of future runs will spend between 5353.825s and 11454.573s for ESA-GA and between 664.035s and 1351.434s, evidencing that the tolerance interval of ESA-GA is much higher than that of mGA.

Although ESA-GA is statistically better than mGA in terms of cost minimization, mGA required much less computation effort and presented a better computation time stability with lower computation gaps. Considering the higher instability of ESA-GA in terms of computation time, mGA seems to be more promising in industrial applications than the ESA-GA, even if it presented a lower efficiency in cost minimization.

## 5.2. Case 2: Modular sneaker

Footwear companies like Adidas and Nike have disseminated the culture of mass customizing shoes on the internet. Anyone can easily test different shoe models that would fit their needs by selecting among different available feature options, which are generally restricted to shoes' colour, logotype, and text engraving.

However, in the example proposed here, the focus is on optimizing the shoes' functionalities, meaning that an optimized shoe option is proposed according to the functionalities required by the customer. Then she/he can select, among the available colours, which one she/he desires in each shoes' part of the proposed shoes. The illustrative example has 28 available product functions, 5 machines (19 machine-configurations), 33 operations and 6 modules (20 module instances) that are detailed in Table 29 (see Appendix C for complementary information).

Figure 25 illustrates a variant of the modular customizable sneaker highlighting their six modules.

Table 29. Available modules and correspondent module instances of sneakers.

Modules		M1	M2	M3	M4	M5	M6
		Insole	Midsole	Outsole	Toe box	Vamp	Heel counter
Module instances	I1	EVA foam	EVA	Natural rubber – Deep groove	Wide	Leather	Cardboard - Straight
	I2	Gel	Polyurethane	Natural rubber - Light	Regular	Synthetic leather	Thermoplastic - semi curved
	I3	Leather	-	Carbon rubber – Deep groove	Narrow	Suede	Without heel counter reinforcement
	I4	-	-	Carbon rubber – Light groove	-	Canvas	-
	I5	-	-	-	-	Mesh	-

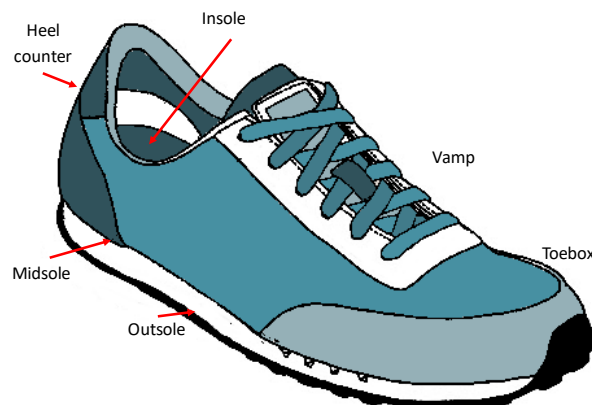


Figure 25. Illustration of a variant of the customizable modular sneaker.

Table 31 summarizes the available functions and the module instances capable of satisfying them. In the example considered here the following functions were selected by the customer: 5, 9, 12, 13, 17, 20, 21, 24, 28.

Table 30. Module instances capable of satisfying each available functional requirement.

<b>Function</b>	<b>Function description</b>	<b>Can be satisfied by module instance(s):</b>
F1	Where: the shoe will be used indoor	M34
F2	Where: the shoe will be used outdoor	M31   M32   M33   M34
F3	Where: the shoe will be used in nature	M31   M32   M33
F4	Weather: looking for a winter shoe	M51   M52   M53
F5	Weather: looking for a summer shoe	M52   M53   M54   M55
F6	Weather: looking for a shoe for run when it is raining	M51   M52
F7	Shoe's weight: does not need a very light shoe	M31   M32   M33   M34
F8	Shoe's weight: prefers a shoe not so heavy neither so light	M31   M32   M33   M34
F9	Shoe's weight: needs a very light shoe	M33   M34
F10	Cushioning: low level	M22
F11	Cushioning: medium level	M21   M22
F12	Cushioning: high level	M21
F13	Sport type: for running	M33   M34
F14	Sport type: for trail running	M33   M34
F15	Sport type: for hiking	M31   M33   M34
F16	Sport type: for walking	M31   M32   M33   M34
F17	Prefers a breathable shoe	M53   M54   M55
F18	Prefers a waterproof shoe	M51   M52
F19	Prefers a water resistant shoe	M51   M52
F20	Prefers a moisture wicking shoe	M52   M53   M54   M55
F21	Prefers an ankle support	M61   M62
F22	Does not need an ankle support	M63
F23	Prefers a narrow show	M43
F24	Prefers a regular shoe	M42
F25	Prefers a wide shoe	M41
F26	Stride type is neutral	M11   M12   M13
F27	Stride type is over pronated	M11   M12
F28	Stride type is pronated	M11   M12

GA parameters were tuned using a full 2-level factorial design planned for 10 replicates ( $10 \times 2^4 = 160$  runs), according to the levels presented in Table 31. The lower and upper level values were defined based on preliminary tests.



Table 31. Evaluated GA parameters and their correspondent tested levels for sneakers' case study.

Parameters	Algorithm	Notation	Levels	
			-1	1
Crossover rate	ESA-GA	$c$	0.7	0.9
Tournament size	Both	$t$	3	7
Population size	Both	$p$	200	600
Number of generations	Both	$n$	200	300
Ration good/ random ind	mGA	$r$	0.2	0.5

The best fitness ( $f_{bf}$ ) and time ( $f_{ct}$ ) functions modelled from the results obtained in the experiments are presented in Table 32.

Table 32. Functions modelled and their  $R^2$  adjusted for mGA and ESA-GA.

Function	ESA-GA	mGA
$f_{bf}$	$0.1043 + 0.0818 c - 0.00068 t - 0.000014 p - 0.000143 n - 0.00849 c * t - 0.000112 c * p + 0.000000 t * p + 0.000027 t * n + 0.000000 p * n + 0.000022 c * t * p - 0.000000 t * p * n$	$0.2662 - 0.226 r - 0.02321 t - 0.000167 p - 0.000323 n + 0.0563 r * t + 0.000284 r * p + 0.000495 r * n + 0.000028 t * p + 0.000078 t * n + 0.000000 p * n - 0.000069 r * t * p - 0.000173 r * t * n - 0.000001 r * p * n - 0.000000 t * p * n + 0.000000 r * t * p * n$
$R^2_{adj}$	62.16%	52.76%
$f_{ct}$	$6148 - 8507 c - 1461 t - 53.7 p - 61.0 n + 1841 c * t + 61.6 c * p + 82.9 c * n + 12.40 t * p + 11.6 t * n + 0.327 p * n - 14.50 c * t * p - 14.6 c * t * n - 0.369 c * p * n - 0.0646 t * p * n + 0.0776 c * t * p * n$	$-67 + 91 r + 25.6 t - 0.689 p - 0.047 n + 0.794 r * p - 0.0421 t * p + 0.00809 p * n$
$R^2_{adj}$	62.94%	86.46%

The optimal parameters of each approach obtained through Minitab's desirability optimization method are presented in Table 33. These values were used to tune each algorithm, which subsequently ran 100 independent times.

Table 33. Optimal GA parameter values for the sneaker's example.

Method	ESA-GA	GA
Crossover rate	0.7	-
Mutation rate	0.3	-
Population size	600	600
Number of generations	200	200
Ration good/random ind.	-	0.2
Tournament size	3	3
Composite desirability (D)	0.7077	0.7950

The results obtained for the example considering the initial parameters values described in Table 34 are summarized in Table 35. The optimal result obtained by both methods corresponds to that presenting the minimum overall manufacturing cost among the 100 runs. The overall minimal time does not necessarily correspond to the time spent by the solutions presenting the minimal cost.

Table 34. Input parameters values.

Required product functions	Initial machines' configurations	Initial layout configuration
3, 5, 9, 12, 13, 17, 20, 21, 24, 28	W1C3, W2C1,W3C1, W4C2, W5C3	W1:L1, W2: L2, W3: L3, W4:L4, W5: L5

Both optimal results suggest that the layout should be reconfigured by switching the positions of machine one (W1) and machine two (W2) with each other, which corresponds to the following layout configuration: W2:L1; W1:L2; W3:L3; W4:L4; W5:L5. Differing from the other case study, here the costs of displacing machines were not much higher than the costs of handling material or reconfiguring machines; therefore, the layout was reconfigured here in order to reduce the material handling costs. This confirms the benefits of concurrently optimizing costs of layout reconfiguration, material handling, and machine-reconfiguration when the machine displacement costs are not much higher than the other costs.

The product variants found by both methods are not the same. They differ within each other only in module 6, since the variant of ESA-GA included module instance M61 (cardboard straight) while mGA included the M62 (thermoplastic semi-curved). All the other module instances selected by both product variants were similar, corresponding to a sneaker containing

EVA foam insole (M11) and midsole (M21), a carbon rubber – deep groove outsole (M33) with a mesh vamp (M55) having a regular toe box (M42).

Table 35. Results obtained for a modular customizable sneaker.

	Method	ESA-GA			mGA		
Statistical parameters for the 100 runs data		Average	Median	Standard deviation	Average	Median	Standard deviation
	Cost (€)	85.3	85.1	3.616	97.6	97.1	7.276
	Comput.. Time (s)	1915.1	1819.6	372.395	560.1	566.3	115.791
		Tolerance interval for 95% of population		Confidence level (%)	Tolerance interval for 95% of population		Confidence level (%)
	Cost (€)	(77.063, 93.506)		95	(81.377, 113.884)		95
	Comput.. Time (s)	(1080.673, 2745.671)		96.3	(354.85, 889.591)		96.3
Optimal results	Cost (€)	Comput.. Time(s)		Cost (€)	Comput. Time(s)		
	77.7	1395.3		80.7	354.9		
Module instances	M1I1, M2I1, M3I3, M4I2, M5I5, <b>M6I1</b>			M1I1, M2I1, M3I3, M4I2, M5I5, <b>M6I2</b>			
Process plan	1:OP27( <b>W4C2</b> )→2:OP21(W4C2)→3:OP2(W4C2)→4: <b>OP31(W4C4)</b> →5:OP20(W3C1)→6: <b>OP4(W3C1)</b> →7:OP17(W3C2)→8: <b>OP28(W3C2)</b> →9: <b>OP3(W3C2)</b> →10:OP30(W3C3)→11: <b>OP5(W3C3)</b> →12:OP24(W3C3)→13:OP14(W1C3)→14:OP8(W1C5)→15:OP25(W1C1)→16:OP12(W4C3)→17: <b>OP18(W4C3)</b> →18: <b>OP13(W5C3)</b> →19:OP6(W5C3)			1:OP14(W1C3)→2:OP2(W4C2)→3:OP21(W4C2)→4: <i>OP33(W4C2)</i> →5: <b>OP31(W5C3)</b> →6: <b>OP4(W5C2)</b> →7: <b>OP12(W5C2)</b> →8: <b>OP5(W5C2)</b> →9: <b>OP27(W5C2)</b> →10: <b>OP13(W5C2)</b> →11:OP20(W3C1)→12:OP17(W3C2)→13:OP24(W3C3)→14:OP30(W3C3)→15:OP8(W1C5)→16: <b>OP28(W1C5)</b> →17: <b>OP18(W1C2)</b> →18: <b>OP3(W1C2)</b> →19:OP6(W1C1)→20:OP25(W1C1)			
Final layout	W2:L1; W1:L2; W3:L3; W4:L4; W5:L5			W2:L1; W1:L2; W3:L3; W4:L4; W5:L5			

Since both product variants have many similar module instances, the only operation differing between them is operation 33 (highlighted in italic), which is only required by module instance M62. Most of the time, a particular operation was performed by the same machine-configuration in both methods' process plans, except those highlighted in bold. The operations' sequence obtained by both methods was very different, backing-up the already known impact of material handling and configuration changes on the overall manufacturing cost.

Despite the differences within both product variants and their process plans, the minimal overall costs attained by both products are quite close. While the product variant from ESA-GA costs 77.7€, the cost of the mGA's product variant is 80.7€. Table 35 shows that we can be 95% confident that at least 95% of future runs of ESA-GA will attain optimal costs within 77.063€

and 93.506€ ( $\Delta=16.47$ ), and mGA will attain costs between 81.377€ and 113.884€ ( $\Delta=32.507\text{€}$ ).

On average, ESA-GA appeared to perform better than mGA with a mean (85.3€) more than 10% lower than the mean of mGA (97.6€), keeping a standard deviation corresponding to the half of the  $\sigma$  presented by mGA. The plot in Figure 26 evidences the better performance of ESA-GA. Since the mean of total cost for ESA-GA is smaller than the mGA and the confidence intervals for their means of total cost do not overlap, the population mean of ESA-GA is probably statistically lower than the mGA.

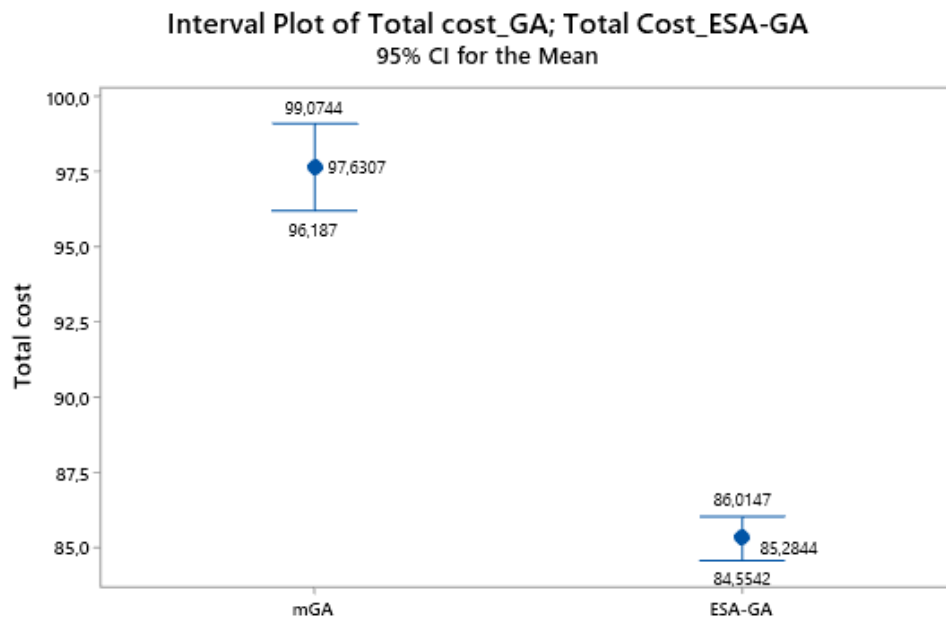


Figure 26 Interval plot of total costs obtained by mGA and ESA-GA in the sneaker case study.

A 2-Sample-t test was carried out to compare the difference of means of both distributions, normally distributed according to the Anderson Darling test (Table 36). The results confirm that the means of ESA-GA and mGA differ by  $|12.246|$  at the 0.05 significance level and we can be 95% confident that this difference will keep between -13.958 and -10.735.

Even though ESA-GA performed better than mGA for minimizing costs, the latter presented a better performance regarding the computation time since it was more than 3 times faster than the ESA-GA. Comparing to the previous case study, the computation times for mGA and ESA-GA were lower here, reduced on average by 37.3% and 75.9%, respectively. The reduction in time for mGA can be explained by the lower number of generations and smaller population size, directly affecting the computation time. For ESA-GA, the population size kept

the same, but the number of generations decreased from 300 to 200. Besides that, the main reason for an expressive reduction of computation time for ESA-GA is that in the example addressed by this case study only 8 product variants can satisfy the required functions, which corresponds to one-third of the number of product variants of the smartphone example.

Table 36. Anderson Darling and 2 Sample t tests for the distributions of total cost obtained by ESA-GA and mGA.

Total cost	Index	Anderson Darling Normality Test
ESA-GA	1	p=0.056
mGA	2	p=0.565
2 Sample t Test		
P-value (p)	Point estimate	CI : 95%
p < 0.05	$\bar{x}_1 - \bar{x}_2 = -12.346$	$\bar{x}_1 - \bar{x}_2 = (-13.958, -10.735.)$

The ESA-GA depends on the number of product variant candidates. The lower the number of product candidates, the faster is the ESA-GA, enabling its application. We can be 96.3% confident that 95% of future runs for ESA-GA will spent a time within 1090.673s (~18.2 min) and 2745.671s (~45.8 min), while the time spent by mGA will keep between 354.85s (~5.8 min) and 889.591s(~14.8 min). Therefore, in this case study the ESA-GA will be able to find a solution in less than one hour, differing from the smartphone case study, in which ESA-GA can spend up to 3.2 hours to find a result.

### 5.3. Analysis and discussion

As previously stated, the integrated approach proposed here focuses on optimizing the product configuration while integrating the individual customer requirements and the RMS configuration. Both illustrative-examples and solution methods have proven that the integrated GA-based approaches can provide good solutions that meet all customer requirements since all required product functions were satisfied.

Further, both methods have proven their applicability for different types of products varying from the technological to the footwear industry. While in the smartphone it presented a one-to-one mapping from functions to modules, meaning that each function was satisfied by a single module instance into the product, in the sneaker case, the one-to-many mapping was used in which a single function was simultaneously satisfied by instances of different modules compounding the product.

Huffman and Kahn (1998) state that presenting the product attributes/functionalities information, instead of product alternatives only, can increase customer satisfaction and reduce the purchase process complexity. The integrated approach proposed here is based on presenting product attributes/functionalities in order to help customers in their decision-making process.

Here customers do not face technical, sometimes incomprehensible, product feature names (GB RAM, system on a chip, etc.) as they usually do in smartphone case, for example. Besides, customers do not choose a sneaker among a set of pre-defined options, which is generally done in the footwear industry. Instead, customers are guided by easy-questions, enabling them to find a customized modular shoe capable of meeting their needs at the lowest cost. This approach is ideal for non-experts or customers new to a certain product category, who do not always have enough knowledge about the product and have difficulties finding what they want (Huffman and Kahn 1998b).

Regarding the questions asked to the customer, they are generic and were formulated to illustrate the examples without considering real customer perception surveys. This is because the investigation concerning how these questions are formulated and/or asked is not in this work's scope. Researchers working on marketing or business management dedicated to MC are more fitted for structuring these questions.

Both examples used here are a simplification of a real-world product. A real sneaker and smartphone have much more production operations and product module instances. Further, to the best of our knowledge, currently, there is still no production system for sneakers or smartphones that is completely reconfigurable, as addressed here. Especially for the sneakers

industry, we are aware that most of them still have many manual process steps. However, these examples' objective was to illustrate how the proposed approach could work for real products and future production systems.

Concerning the algorithm's performance, in both case studies the means obtained by both methods were statistically different, with ESA-GA attaining lower means of costs than mGA. Hence, ESA-GA outperformed mGA in both case studies in terms of cost minimization. A possible reason for mGA presenting higher means of cost and higher standard deviation than ESA-GA is that in mGA all decisions are made simultaneously, including product module selection and required operations identification. In contrast, ESA-GA selects in the first step (ESA) all possible product variants and the required operations; therefore, on the second step (GA), only the operations sequence and their assignment, as well as the layout configuration, are chosen. This reduces the solution space for the ESA-GA method at the GA level (step 2) and may contribute to reducing solution gaps since the probability of the algorithm being trapped into bad solutions (far from the optimum) is smaller.

The shoe case study was used to compare the evolution process of ESA-GA and mGA (Figure 27). The total cost data corresponds to the average over 50 replicates and it is represented in logarithmic scale for an easy visualization of the convergence process. As explained in Chapter 3, the penalty strategy was used to handle some constraints. Since mGA englobes the product variant selection into the evolution process, it has more constraints to be handled, implying higher penalty values and consequently the fitness values are higher. This may explain why mGA starts with higher values of total cost than ESA-GA.

Both algorithms rapidly reduce the average costs at the beginning of the evolutionary process. The convergence of ESA-GA appears to be between generations 125 and 150, while for mGA the convergence appears to be within generations 150 and 175, and no more considerable improvement was observed afterwards. The evolutionary plot of Figure 27 shows that ESA-GA attains lower average costs than mGA, confirming the results obtained with the statistical analyses previously presented.

The mGA was much faster than ESA-GA in both illustrative examples presenting much lower computation time gaps. The average time of mGA in the smartphone example was about double of the shoe example. This could be mainly related to the number of generations and population size that were higher for the smartphone example.

Due to its design, ESA-GA computation time is very sensitive to the number of product variants capable of satisfying the set of required product functions. The number of product

variants for smartphone case (24) is higher than the shoe case (8). Besides that, the number of generations in the shoe case study (200) was lower than the number of generations used in the smartphone case study (300).

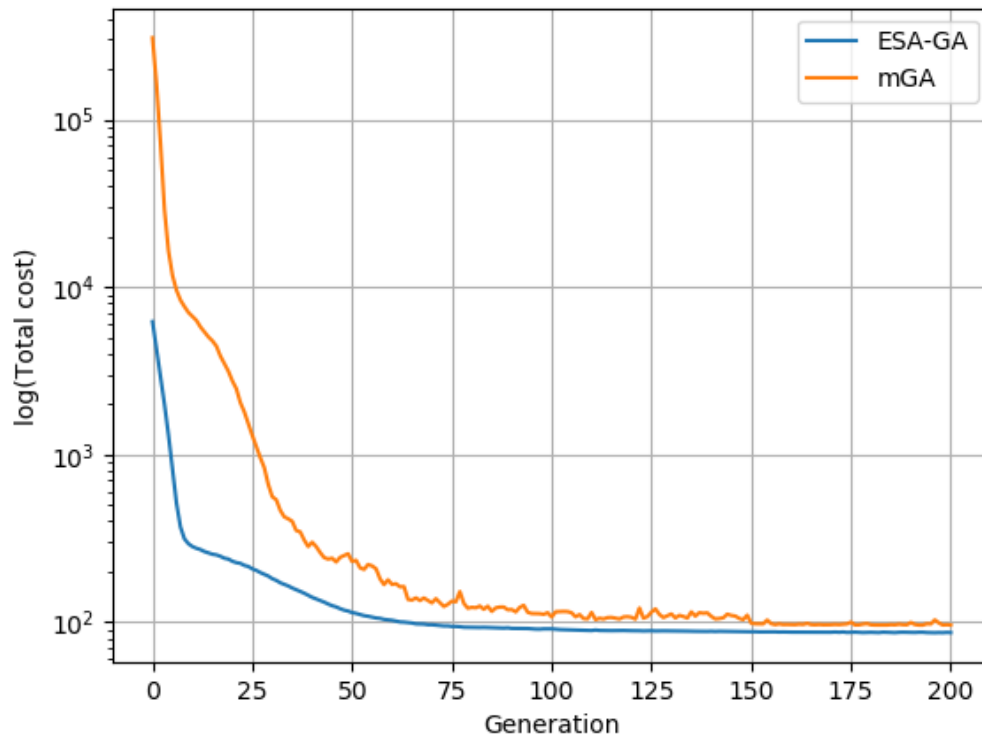


Figure 27. Convergence process of  $\log(\text{Total cost})$  for mGA (orange line) and ESA-GA (blue line).

On average, ESA-GA takes about 31.9 minutes and 2.2 hours to optimize the shoe and smartphone case studies, respectively. In contrast, mGA spends, on average, 9.3 minutes and 15.4 minutes for solving the shoe and smartphone examples, respectively. Although ESA-GA outperforms mGA in cost minimization, mGA appears to be more promising for practical applications because it required much less computation time than ESA-GA in both examples. Besides that, the computation time of mGA for both case studies was more stable, since the average time required by this method varied about 6.1 minutes within both examples while for ESA-GA this variation was almost 1.7 hours.

Even though the computation times spent by mGA are relatively low due to the problems' sizes addressed in this chapter, in practical terms they are still too long. Considering both case studies, the mGA method application appears to be more promising in configurators for technological devices, when the customer is unfamiliar with the technical features of the product and do not really know which features will better fit their needs. Considering the current



configurators for sneakers, the mGA method can only be applied to optimize configurators for this type of product if some improvements in the algorithm are made in order to reduce the computation time.

Since this work was mainly focused on integrating the configuration decisions from product to RMS, the two solution approaches proposed were based on GA due to its wide applicability and robustness to solve optimization problems related to modular products and RMS configurations. Both GA-based approaches were modified to adapt to the problem addressed here, differing from the standard GA.

Some improvements were made in the algorithms at the initialization level (mGA) and at the evolutionary operators (both). For future research, other improvements can be considered, such as coupling the mGA with local search techniques, such as simulated annealing and tabu search. Further, other metaheuristics based on evolutionary computation (e.g., differential evolution, evolutionary strategy) or in swarm intelligence (e.g., particle swarm or ant colony optimization) can also be developed, and their performance can be compared to that of the methods proposed here.

## GENERAL CONCLUSION

### Main contributions

More companies are implementing mass customization (MC) to meet the uniqueness of each individual customer's requirements. It is no more enough to mass-customize to attain a market segment. The focus now is on meeting each specific customer requirement, and to do that, it is important to integrate customers into the product development decisions.

Products and manufacturing processes have also evolved to adapt to the new market demands. Products and processes have increasingly become modular and reconfigurable. At the product level, modularity and reconfigurability allow increasing the product variety since a large set of product options can be generated from a relatively small set of components. At the process level, the RMS are currently the most modular and reconfigurable processes. They are known as the most appropriate to deal with high variety demand and market instability due to their ability to be rapidly reconfigured at the machine and layout levels, changing their functionality and capacity (Koren 2010). Further, the reconfigurability of RMS at the machine level has been widely explored in the literature. However, there are still few works addressing the RMS layout design issues, including layout reconfiguration and costs of material handling.

Decisions of product and process design are highly interrelated, and to take advantage of the benefits provided by the modular design, it is important to concurrently analyze product and process performances. Further, in MC contexts, the specific customer needs should also be considered when selecting the product configuration. However, there are still very few works concurrently optimizing decisions of modular product and RMS for MC, driven by individual customer requirements. Moreover, no work concurrently optimized the product configuration, driven by individual customer requirements, concurrently to the process planning and RMS layout configuration for MC.

Considering the previous statements, it seems relevant to integrate product and RMS decisions for mass customizing products according to specific customer requirements. This is why this work proposed two approaches to concurrently optimize the configuration of modular products and RMS driven by individual customer requirements:

- 1) Concurrent optimization of product configuration, driven by individual customer requirements, and process planning in an RMS to minimize the overall manufacturing cost (modeled as a Integer Linear Programming model - ILP);

- 2) Concurrent optimization of product configuration, driven by individual customer requirements, and process planning and layout configuration in an RMS to minimize the overall manufacturing cost (modeled as a Nonlinear Integer Programming - NLIP).

This work proposes the concurrent optimization of product configuration and process planning in RMS, considering that a given module instance can be manufactured in different machine-configurations instead of fixing a given RMS configuration per module instance. Since in RMS, the same operation can be performed by different machine-configurations and a single machine-configuration can perform various operations, establishing a single RMS configuration for each module instance would reduce the benefits of RMS modularity and reconfigurability.

Both mathematical models proposed in this work were validated by means of a small case study, which was solved by deterministic approaches. Some analyses were carried out to check the impact of demanding different product functions as well as the impact of starting with different configurations of machines in a given layout. The impact of layout reconfiguration costs was also investigated.

Two solution approaches based on evolutionary algorithms were proposed to solve the mathematical model for higher instances. Both methods were used to solve the small case study, and their results were compared to those obtained by the deterministic approaches in order to validate them. Once they were validated, both approaches were used to solve two illustrative examples based on real products, presenting a higher number of instances.

The results found in this work prove that:

- There are benefits from concurrently minimizing costs of product configuration with the RMS configuration for meeting specific customer requirements. Depending on the required functions and the current machines' configurations, different optimal product configurations will attain minimal overall manufacturing costs;
- The costs of displacing machines can highly affect the layout reconfiguration. The results obtained in this work suggest that the layout should be reconfigured only when the new machines placement can considerably reduce the material handling costs, compensating the additional costs of reconfiguring the layout. This trade-off generally occurs when displacing machines' costs are not substantially higher than the costs of handling material and changing machines' configurations. Therefore, when machines

cannot be easily displaced, it is preferable to concurrently optimize the product configuration with the process planning only, by using the ILP model that requires less computation effort compared to the NLIP one;

- The initial configuration of machines can greatly impact the overall manufacturing production cost for configuring a product according to a set of functions. The total production cost varies about 30% considering all initial machines' configurations in a given layout for the considered example;
- Both Genetic Algorithm-based solution approaches have proven their ability to solve the concurrent optimization of product configuration with the process planning and layout configuration in an RMS for problems presenting mid-size instances. The ESA-GA presented a better performance in terms of cost minimization, while mGA ensured lower and stable computation times.
- Due to the complexity involved in this optimization problem, it is possible to affirm that both methods solved the NLIP problem in relatively low time, spending less than 3.5 hours (ESA-GA) or up to 22 minutes (mGA). However, considering practical applications, some improvements into the algorithms are still required for reducing their computational effort.
- Due to the mGA stability in computation time, it appears to be more promising for industrial applications. However, some improvements are still required to reduce its computation effort, making possible its application for solving high-sized problems within less time;
- The solution approaches are capable of solving configuration problems for different types of products, varying from technological to footwear industry. However, the application in technological industry may be more promising considering the higher probability of customer being unfamiliar with technical features, requiring help in terms of product functionalities;

### **Research limitations and future work directions**

Making all decisions simultaneously considerably increases the problem complexity. The results provided by both GA-based approaches suggested that dividing the problem into more steps (ESA-GA) can help to get better fitness but will require a higher computation effort, especially due to the exhaustive search method. When the mGA was used singly, even with an

heuristic initialization, it was not capable of attaining costs as low as those obtained by the ESA-GA.

Therefore, to overcome the disadvantages of the methods presented in this work, other metaheuristic methods can be proposed to improve the algorithm's ability to find better solutions within lower computation time. A possibility would be using a hybrid method coupling a local search algorithm, such as the hill climbing, with a population-based algorithm, such as GA, which can contribute to improving the exploration and exploitation search phases.

Both propositions consist of uni-objective methods, in which only the cost is minimized. However, other performance criteria than costs can affect the final decisions; therefore, future work can develop mathematical models to optimize multi-objective problems involving product and process configuration issues. Considering the increasing interest in sustainable manufacturing, a possible performance criterion to be concurrently optimized with costs is the greenhouse gas emissions of product and process. Since products are optimized according to specific customer requirements, another possible performance criterion (response of interest) would be the customer satisfaction index.

This work proposes an approach to guide the customers during their choice with a promising capacity to help them finding out what they really want, reducing the “mass confusion” paradigm. Nevertheless, it was still not applied to a real case to evaluate the customer experience during the customization process. Hence, in future work, the optimization algorithms proposed here will be integrated into an online configurator (MC toolkit) of shoes to investigate the customers' expectations and measure their satisfaction level, through a satisfaction survey. The feedback obtained from customers can be posteriorly used to extract personal customer needs through natural language processing. The information obtained from that can subsequently be applied to improve the performance of the optimization methods as well as the configurator structure (i.e., the way questions are asked, words used in the questions, etc.).

## BIBLIOGRAPHY

- Abbasi M, Houshmand M (2011) Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. *International Journal of Advanced Manufacturing Technology* 54(1–4):373–392
- Abbasi M, Houshmand M (2009) Production Planning of Reconfigurable Manufacturing Systems with Stochastic Demands Using Tabu Search. *International Journal of Manufacturing Technology and Management* 17(1–2):125–148
- Abdi MR, Labib A (2016) RMS capacity utilisation: product family and supply chain. *International Journal of Production Research* 55(7):1930–1956
- Allen KR, Carlson-Skalak S (1998) Defining product architecture during conceptual design. In: ASME design engineering technical conference, DETC98/DTM-5650, Atlanta, GA
- Asghar E, Zaman UK, Baqai AA, Homri L (2018) Optimum machine capabilities for reconfigurable manufacturing systems. *International Journal of Advanced Manufacturing Technology* 95(9–12):4397–4417
- Ashraf M, Hasan F (2018) Configuration selection for a reconfigurable manufacturing flow line involving part production with operation constraints. *International Journal of Advanced Manufacturing Technology* 98(5–8):2137–2156
- Baldwin CY., Clark KB (1997) Managing in an age of modularity. *Harvard Business Review* 75:84–93
- Battaïa O, Dolgui A, Guschinsky N (2016) Decision support for design of reconfigurable rotary machining systems for family part production. *International Journal of Production Research* 55(5):1368–1385
- Benderbal HH, Benyoucef L (2019) Machine layout design problem under product family evolution in reconfigurable manufacturing environment: a two-phase-based AMOSA approach. *International Journal of Advanced Manufacturing Technology* 104(1–4):375–389
- Benderbal HH, 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
- Benderbal HH, Dahane M, Benyoucef L (2018) Modularity assessment in reconfigurable manufacturing system (RMS) design: an Archived Multi-Objective Simulated Annealing-based approach. *International Journal Advanced Manufacturing Technology* 94:729–749
- Bensmaine A, Dahane M, Benyoucef L (2013) A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers and Industrial Engineering* 66(3):519–524
- Bi ZM, Lang SYT, Shen W, Wang L (2008) Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research* 46(4):967–992
- Blum C, Roli A (2003) *Metaheuristics in Combinatorial Optimization: Overview and Conceptual*

Comparison. *ACM Computing Surveys* 35(3):268–308

- Bortolini M, Galizia FG, Mora C, Pilati F (2019) Reconfigurability in cellular manufacturing systems: a design model and multi-scenario analysis. *International Journal of Advanced Manufacturing Technology* 104(9–12):4387–4397
- Bryan A, Hu SJ, Koren Y (2013) Assembly System Reconfiguration Planning. *Journal of Manufacturing Science and Engineering* 135(4):1–14
- Bryan A, Ko J, Arbor A (2007) Co-Evolution of Product Families and Assembly Systems. *CIRP Annals - Manufacturing Technology* 56(2):41–44
- Burke EK, Newall JP, Weare RF (1998) Initialization Evolutionary Strategies and Timetabling Diversity in. *Evolutionary Computation* 6(1):81–103
- Cabigiosu A, Camuffo A (2017) Measuring Modularity: Engineering and Management Effects of Different Approaches. *IEEE Transactions on Engineering Management* 64(1):103–114
- Chaves AA, Gonçalves JF, Lorena LAN (2018) Adaptive biased random-key genetic algorithm with local search for the capacitated centered clustering problem. *Computers and Industrial Engineering* 124:331–346
- Chen L-C, Lin L (2002) Optimization of product configuration design using functional requirements and constraints. *Research in Engineering Design* 13:167–182
- Chen Z, Wang L (2010) Personalized product configuration rules with dual formulations: A method to proactively leverage mass confusion. *Expert Systems with Applications* 37:383–392
- Choi Y-C, Xirouchakis P (2015) A holistic production planning approach in a reconfigurable manufacturing system with energy consumption and environmental effects. *International Journal of Computer Integrated Manufacturing* 28(4):379–394
- Da Cunha C, Agard B, Kusiak A (2010) Selection of modules for mass customisation. *International Journal of Production Research* 48(5):1439–1454
- Da Cunha C, Agard B, Kusiak A (2007) Design for Cost: Module-Based Mass Customization. *IEEE Transactions on Automation Science and Engineering* 4(3):350–359
- Daaboul J, Bernard A, Laroche F (2011) Knowledge Management for Mass Customisation. In: *Global Product Development*. pp 421–426
- Davis S (1987) *Future Perfect*. Addison-Wesley Publishing, Reading, MA
- Davrajh S, Bright G (2013) Advanced quality management system for product families in mass customization and reconfigurable manufacturing. *Assembly Automation* 33(2):127–138
- Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18(4):577–601
- Dou J, Dai X, Meng Z (2009a) Precedence graph-oriented approach to optimise single-product flow-line configurations of reconfigurable manufacturing system. *International Journal of Computer*

- Dou J, Dai X, Meng Z (2010) Optimisation for multi-part flow-line configuration of reconfigurable manufacturing system using GA. *International Journal of Production Research* 48(14):4071–4100
- 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):301–317
- Dou J, Dai X, Meng Z (2009b) Graph theory-based approach to optimize single-product flow-line configurations of RMS. *International Journal of Advanced Manufacturing Technology* 41(9–10):916–931
- Dou J, Li J, Su C (2016) Bi-objective optimization of integrating configuration generation and scheduling for reconfigurable flow lines using NSGA-II. *International Journal of Advanced Manufacturing Technology* 86(5–8):1945–1962
- Dou J, Li J, Xia D, Zhao X (2020a) A multi-objective particle swarm optimisation for integrated configuration design and scheduling in reconfigurable manufacturing system. *International Journal of Production Research*. <https://doi.org/https://doi.org/10.1080/00207543.2020.1756507>
- Dou J, Su C, Zhao X (2020b) Mixed integer programming models for concurrent configuration design and scheduling in a reconfigurable manufacturing system. *Concurrent Engineering Research and Applications* 28(1):32–46
- Dou R, Zong C (2014) Application of Interactive Genetic Algorithm based on hesitancy degree in product configuration for customer requirement. *International Journal of Computational Intelligence Systems* 7(SUPPL.2):74–84
- Drira A, Pierreval H, Hajri-Gabouj S (2007) Facility layout problems: A survey. *Annual Reviews in Control* 31(2):255–267
- Eguia I, Molina JC, Lozano S, Racero J (2017) Cell design and multi-period machine loading in cellular reconfigurable manufacturing systems with alternative routing. *International Journal of Production Research* 55(10):2775–2790
- ElMaraghy H, Schuh G, Elmaraghy W, Piller F, Schönsleben P, Tseng M, Bernard A (2013) Product variety management. *CIRP Annals - Manufacturing Technology* 62(2):629–652
- ElMaraghy HA (2007) Reconfigurable Process Plans For Responsive Manufacturing Systems. In: *Digital Enterprise Technology*. Boston, MA, pp 35–44
- Feitzinger E, Lee HL (1997) Mass Customization at Hewlett-Packard: The Power of Postponement. *Harvard Business Review* 75:116–121
- Franke N, Piller F, Piller FT (2004) Toolkits for User Innovation and Design: an Exploration of User Interaction and Value Creation in the Watch Market. *Journal of Product Innovation Management* 21(6):401–415
- Franke N, Piller FT (2002) Configuration Toolkits for Mass Customization Setting a Research Agenda. *Innovation* 33(33):1–29
- Gandomi AH., Yang XS., Talatahari S., Alavi AH (2013) Metaheuristic applications in structures and infrastructures. *Newnes*



- Gershenson JK, Prasad GJ, Allamneni S (1999) Modular Product Design : A Life-cycle View. *Journal of Integrated Design and Process Science* 3(4):1–9
- Gershenson JK, Prasad GJ, Zhang Y (2003) Product modularity: Definitions and benefits. *Journal of Engineering Design* 14(3):295–313
- Ghanei S, Algeddawy T (2020) An Integrated Multi-Period Layout Planning and Scheduling Model for Sustainable Reconfigurable Manufacturing Systems. *Journal of Advanced Manufacturing Systems* 19(1):31–64
- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company
- Goldberg DE, Deb K (1991) A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. Morgan Kaufmann Publishers, Inc.
- Goldberg DE, Deb K, Korb B (1990) Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale. *Complex Systems* 4:415–444
- Goldberg DE, Korb B, Deb K (1989) Messy Genetic Algorithms: Motivation, Analysis, and First Results. *Complex Systems* 3:493–530
- Goyal KK, Jain PK (2016) Design of reconfigurable flow lines using MOPSO and maximum deviation theory. *International Journal of Advanced Manufacturing Technology* 84(5–8):1587–1600
- Goyal KK, Jain PK, Jain M (2013) Applying Swarm intelligence to design the reconfigurable flow lines. *International Journal of Simulation Modelling* 12(1):17–26
- Goyal KK, Jain PK, Jain M (2012) Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS. *International Journal of Production Research* 50(15):4175–4191
- Guan X, Dai X, Qiu B, Li J (2012) A revised electromagnetism-like mechanism for layout design of reconfigurable manufacturing system. *Computers and Industrial Engineering* 63(1):98–108
- Gupta A, Jain PK, Kumar D (2015) Configuration selection of reconfigurable manufacturing system based on performance. *International Journal of Industrial and Systems Engineering* 20(2):209–229
- Gupta A, Jain PK, Kumar D (2014) A novel approach for part family formation for reconfiguration manufacturing system. *Opsearch* 51(1):76–97
- Hassan N, Davrajh S, Bright G (2011) Prioritization of part scheduling with modular quality control in hybrid manufacturing cells for mass customization. In: *IEEE Africon 2011 - The Falls Resort and Conference Centre, Livingstone*. IEEE, Zambia, pp 1–6
- Hoek RI van, Weken HAM (1998) The Impact of Modular Production on the Dynamics of Supply Chains. *The International Journal of Logistics Management* 9(2):33–50
- Holland J (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI
- Höltkä-Otto K, de Weck O (2007) Degree of modularity in engineering systems and products with

- technical and business constraints. *Concurrent Engineering Research and Applications* 15(2):113–125
- Hong G, Hu L, Xue D, Tu YL, Xiong YL (2008) Identification of the optimal product configuration and parameters based on individual customer requirements on performance and costs in one-of-a-kind production. *International Journal of Production Research* 46(12):3297–3326
- Huffman C, Kahn BE (1998a) Variety for sale: Mass customization or mass confusion? *Journal of Retailing* 74(4):491–513
- Huffman C, Kahn BE (1998b) Variety for sale: Mass customization or mass confusion? *Journal of Retailing* 74(4):491–513
- Jacobs M, Droge C, Vickery SK, Calantone R (2011) Product and Process Modularity's Effects on Manufacturing Agility and Firm Growth Performance. *Journal of Product Innovation Management* 28(1):123–137
- Jacobs M, Vickery SK, Droge C (2007) The effects of product modularity on competitive performance. *International Journal of Operations & Production Management* 27(10):1046–1068
- Jannach D, Zanker M (2013) Modeling and solving distributed configuration problems: A CSP-based approach. *IEEE Transactions on Knowledge and Data Engineering* 25(3):603–618
- Jiao J, Simpson TW, Siddique Z (2007) Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing* 18(1):5–29
- Jiao J, Tseng MM (1999) Methodology of developing product family architecture for mass customization. *Journal of Intelligent Manufacturing* 10(1):3–20
- Jiao J, Tseng MM (2000) Fundamentals of product family architecture. *Integrated Manufacturing Systems* 11(7):469–483
- Jiao Y, Yang Y (2019) A product configuration approach based on online data. *Journal of Intelligent Manufacturing* 30(6):2473–2487
- Khalaf REH, Agard B, Penz B (2011a) Simultaneous design of a product family and its related supply chain using a Tabu Search algorithm. *International Journal of Production Research* 49(19):5637–5656
- Khalaf REH, Agard B, Penz B (2011b) Module Selection and Supply Chain Optimization for Customized Product Families Using Redundancy and Standardization. *IEEE Transactions On Automation Science And Engineering* 8(1):118–129
- Khalaf REH, Agard B, Penz B (2010a) An experimental study for the selection of modules and facilities in a mass customization context. *Journal of Intelligent Manufacturing* 21:703–716
- Khalaf REH, Agard B, Penz B (2010b) An experimental study for the selection of modules and facilities in a mass customization context. *Journal of Intelligent Manufacturing* 21(6):703–716
- Khezri A, Benderbal HH, Benyoucef L (2020) Towards a sustainable reconfigurable manufacturing system (SRMS): multi-objective based approaches for process plan generation problem. *International Journal of Production Research*. <https://doi.org/https://doi.org/00207543.2020.1766719>

- Koren Y (2010) *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*. John Wiley & Sons, Inc.
- Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G, Van Brussel H (1999) Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology* 48(2):527–540
- Koren Y, Shpitalni M (2010) Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 29(4):130–141
- Koren Y, Wang W, Gu X (2016) Value creation through design for scalability of reconfigurable manufacturing systems. *International Journal of Production Research* 55(5):1227–1242
- Kristjansdottir K, Shafiee S, Hvam L, Forza C, Mortensen NH (2018) The main challenges for manufacturing companies in implementing and utilizing configurators. *Computers in Industry* 100:196–211
- Kunz J, Smith I, Tomiyama T (2009) A constraint satisfaction approach to resolving product configuration conflict. *Advanced Engineering Informatics* 23(2):139
- Kusiak A (2002) Integrated product and process design: A modularity perspective. *Journal of Engineering Design* 13(3):223–231
- Kusiak A, Smith MR, Song Z (2007) Planning product configurations based on sales data. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 37(4):602–609
- Le Duigou J, Bernard A, Perry N, Delplace JC (2009) Global approach for technical data management. Application to ship equipment part families. *CIRP Journal of Manufacturing Science and Technology* 1(3):185–190
- Lee CH, Chen CH, Lin C, Li F, Zhao X (2019) Developing a quick response product configuration system under industry 4.0 based on customer requirement modelling and optimization method. *Applied Sciences (Switzerland)* 9(23):1–27
- Lee WB, Lau H, Liu Z, Tam S (2001) A fuzzy analytic hierarchy process approach in modular. *Expert Systems* 18(1):32–42
- Li B, Chen L, Huang Z, Zhong Y (2006) Product configuration optimization using a multiobjective genetic algorithm. *International Journal of Advanced Manufacturing Technology* 30(1–2):20–29
- Li J, Dai X, Meng Z, Dou J, Guan X (2009) Rapid design and reconfiguration of Petri net models for reconfigurable manufacturing cells with improved net rewriting systems and activity diagrams. *Computers and Industrial Engineering* 57(4):1431–1451
- Li WD, McMahon CA (2007) A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing* 20(1):80–95
- Liu M, An L, Zhang J, Chu F, Chu C (2019) Energy-oriented bi-objective optimisation for a multi-module reconfigurable manufacturing system. *International Journal of Production Research* 57(19):5974–5995
- Liu X, Du G, Jiao RJ (2017) Bilevel joint optimisation for product family architecting considering make-or-buy decisions. *International Journal of Production Research* 55(20):5916–5941

- Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T (2007) A survey for the quadratic assignment problem. *European Journal of Operational Research* 176(2):657–690
- Maganha I, Silva C, Ferreira LMDF (2019) The layout design in reconfigurable manufacturing systems: a literature review. *The International Journal of Advanced Manufacturing Technology* 105:683–700
- Malhotra R, Singh N, Singh Y (2011) Genetic Algorithms: Concepts, Design for Optimization of Process Controllers. *Computer and Information Science* 4(2):39–54
- Massimi E, Khezri A, Benderbal HH, Benyoucef L (2020) A heuristic-based non-linear mixed integer approach for optimizing modularity and integrability in a sustainable reconfigurable manufacturing environment. *International Journal of Advanced Manufacturing Technology* 108(7–8):1997–2020
- McDermott J (1982) R1: A rule-based configurator of computer systems. *Artificial Intelligence* 19(1):39–88
- Meng X, Jiang Z, Huang G (2007) On the module identification for product family development. *International Journal of Advanced Manufacturing Technology* 35(1–2):26–40
- Mervyn F, Kumar AS, Bok SH, Nee AYC (2004) Developing distributed applications for integrated product and process design. *CAD Computer Aided Design* 36(8):679–689
- Michalos G, Makris S, Mourtzis D (2012) An intelligent search algorithm-based method to derive assembly line design alternatives. *International Journal of Computer Integrated Manufacturing* 25(3):211–229
- Mikkola JH, Gassmann O (2003) Managing modularity of product architectures: Toward an integrated theory. *Transactions on Engineering Management* 50(2):1–15
- Mittal KK, Jain PK, Kumar D (2018) Optimal selection of modular products in reconfigurable manufacturing systems using analytic hierarchy process. *International Journal of Logistics Systems and Management* 30(4):521–537
- Mittal S, Frayman F (1989) Towards a generic model of configuration tasks. In: *Proc. 11th Int'l Joint Conf. Artificial Intelligence*. Morgan Kaufmann, San Francisco, 1989, pp 1395–1401
- Moghaddam SK, Houshmand M, Saitou K, Fatahi Valilai O (2019) Configuration design of scalable reconfigurable manufacturing systems for part family. *International Journal of Production Research* 58(10):2974–2996
- Moghaddam SK, Houshmand M, Valilai OF (2018) Configuration design in scalable reconfigurable manufacturing systems (RMS); a case of single-product flow line (SPFL). *International Journal of Production Research* 56(11):3932–3954
- Mohapatra P, Benyoucef L, Tiwari MK (2013) Integration of process planning and scheduling through adaptive setup planning: a multi-objective approach. *International Journal of Production Research* 51(23–24):7190–7280
- Müller R, Hörauf L, Vette M, Speicher C (2016) Planning and Developing Cyber-physical Assembly Systems by Connecting Virtual and Real Worlds. *Procedia CIRP* 52:35–40

- Musharavati F, Hamouda ASM (2012) Enhanced simulated-annealing-based algorithms and their applications to process planning in reconfigurable manufacturing systems. *Advances in Engineering Software* 45(1):80–90
- Najid N-M, Castagna P, Kouiss K (2020) System Engineering-Based Methodology to Design Reconfigurable Manufacturing Systems. In: Benyoucef L (ed) *Reconfigurable Manufacturing Systems : From Design to Implementation*. Springer Nature Switzerland AG, p 258
- Nallakumarasamy G, Srinivasan P, Raja KV, Malayalamurthi R (2011) Optimization of Operation Sequencing in CAPP Using Superhybrid Genetic Algorithms-Simulated Annealing Technique. *ISRN Mechanical Engineering* 2011(1):1–7
- Newcomb PJ, Bras B, Rosen DW (1998) Implications of Modularity on Product Design for the Life Cycle. *Journal of Mechanical Design* 120(3):483–490
- Ortiz F, Simpson JR, Pignatiello JJ, Heredia-Langner A (2004) A genetic algorithm approach to multiple-response optimization. *Journal of Quality Technology* 36(4):432–450
- Paes IF, Sabioni RC, Daaboul J, Duigou J Le (2018) Product and Process Modular Design : a Review. In: 8th International Conference on Mass Customization and Personalization – Community of Europe, MCP-CE 2018. University of Novi Sad – Faculty of Technical Sciences Department of Industrial Engineering and Management 21000, Novi Sad, Trg Dositeja Obradovića 6, Serbia, pp 97–106
- Pahl G, Beitz W (1995) *Engineering Design: a systematic approach*. Springer Science & Business Media
- Papadimitrou CH, Steiglitz K (1998) *Combinatorial optimization: algorithms and complexity*. Courier Corporation
- Pattanaik LN, Jena A (2018) Tri-objective optimisation of mixed model reconfigurable assembly system for modular products. *International Journal of Computer Integrated Manufacturing* 32(1):72–82
- Pil FK, Cohen SK (2006) Modularity: Implications for imitation, innovation, and sustained advantage. *Academy of Management Review* 31(4):995–1011
- Piller F, Walcher D (2017) *Leading Mass Customization and Personalization: How to profit from service and product customization in e-commerce and beyond*. Amazon Kindle ebook, Seattle
- Piller FT, Blazek P (2014) Core Capabilities of Sustainable Mass Customization. In: *Knowledge-Based Configuration: From Research to Business Cases*. Morgan Kaufmann Publishers, Waltham, MA, pp 107–120
- Pimmler TU, Eppinger SD (1994) Integration analysis of product decompositions. *ASME Design Theory and Methodology Conference September 1994, Minneapolis (September 1994)*:343–351
- Pine BJ (1993a) *Mass customization: The new frontier in business competition*. Harvard Business School Press
- Pine BJ (1993b) Mass customizing products and services. *Planning Review* 21(4):6–55
- Pitiot P, Aldanondo M, Vareilles E (2014) Concurrent product configuration and process planning: Some optimization experimental results. *Computers in Industry* 65(4):610–621

- Pitiot P, Aldanondo M, Vareilles E, Gaborit P, Djefel M, Carbonnel S (2013) Concurrent product configuration and process planning, towards an approach combining interactivity and optimality. *International Journal of Production Research* 51(2):524–541
- Pitiot P, Monge LG, Aldanondo M, Vareilles E, Gaborit P (2020) Optimisation of the concurrent product and process configuration: an approach to reduce computation time with an experimental evaluation. *International Journal of Production Research* 58(2):631–647
- Reed P, Minsker B, Goldberg DE (2000) Designing a competent simple genetic algorithm for search and optimization. *Water Resources Research* 36(12):3757–3761
- Sabin D, Weigel R (1998) Product configuration frameworks—a survey. *IEEE Intelligent Systems and their Applications* 13(4):42–49
- Sahni S, Gonzalez T (1976) P-Complete Approximation Problems. *Journal of the ACM (JACM)* 23(3):555–565
- Salvador F (2007) Toward a product system modularity construct: Literature review and reconceptualization. *IEEE Transactions on Engineering Management* 54(2):219–240
- Sarma J, Jong K De (1995) Generation gap methods. In: *Handbook for Institute of Physics Publishing*. IOP Publishing Ltd, pp 205–211
- Saxena LK, Jain PK (2012) A model and optimisation approach for reconfigurable manufacturing system configuration design. *International Journal of Production Research* 50(12):3359–3381
- Schilling MA (2000) Toward a General Modular Systems Theory and its application to Inferfirm Product Modularity. *Academy of Management Review* 25(2):312–334
- Schilling MA, Steensma HK (2001) The use of modular organizational forms: An industry-level analysis. *Academy of Management Journal* 44(6):1149–1168
- Shaik AM, Rao VVSK, Rao CS (2015) Development of modular manufacturing systems—a review. *International Journal of Advanced Manufacturing Technology* 76(5–8):789–802
- Shamsuzzoha A, Al-kind M, Al-hinai N (2017) Application of Product Modularity in Industry: a Case Study. In: *International Conference on Industrial Engineering and Operations Management*. pp 1936–1945
- Singh PP, Madan J, Singh H (2020) Composite performance metric for product flow configuration selection of reconfigurable manufacturing system (RMS). *International Journal of Production Research*. <https://doi.org/https://doi.org/00207543.2020.1756511>
- Song Q, Ni Y, Ralescu DA (2020) The impact of lead-time uncertainty in product configuration. *International Journal of Production Research* 0(0):1–23. <https://doi.org/10.1080/00207543.2020.1715506>
- Song Z, Kusiak A (2009) Optimising product configurations with a data-mining approach. *International Journal of Production Research* 47(7):1733–1751
- Srinivas M, Patnaik LM (1994) Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics* 24(4):656–667

- Starr MK (1965) Modular-production: a new concept. *Harvard Business Review* 43(6):131–142
- Suh NP (1990) *The Principles of Design*. Oxford University Press Inc.
- Tang D, Wang Q, Ullah I (2017) Optimisation of product configuration in consideration of customer satisfaction and low carbon. *International Journal of Production Research* 55(12):3349–3373
- Tong Y, Tang Z, Mei S, Qiao L, Gong Z (2015) Research on customer-oriented optimal configuration of product scheme based on Pareto genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 229(1):148–156
- Touzout FA, 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 FA, 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
- Trentin A, Perin E, Forza C (2012) Product configurator impact on product quality. *International Journal of Production Economics* 135(2):850–859
- Tseng MM, Wang Y, Jiao RJ (2017) Mass Customization. In: *CIRP Encyclopedia of Production Engineering*. pp 1–8
- Tu Q, Vonderembse MA, Ragu-Nathan TS, Ragu-Nathan B (2004) Measuring modularity-based manufacturing practices and their impact on mass customization capability: A customer-driven perspective. *Decision Sciences* 35(2):147–168
- Ulrich KT (1994) Fundamentals of Product Modularity. In: *Management of Design*. Springer, Dordrecht, pp 219–231
- Ulrich KT (1995) The role of product architecture in the manufacturing firm. *Research Policy* 24(3):419–440
- Ulrich KT, Eppinger SD (2009) *Product Design and Development*. Tata McGraw-Hill Education.
- US Department of Defense (1996) *Guide to Integrated Product and Process Development (Version 1.0)*. WASHINGTON, DC
- Vickery SK, Dro C, Calantone R (2016) Product Modularity , Process Modularity , and New Product Introduction Performance : Does Complexity. *Production and Operations Management* 25(4):751–770
- Voordijk H, Meijboom B, de Haan J (2006) Modularity in supply chains: a multiple case study in the construction industry. *International Journal of Operations & Production Management* 26(6):600–618
- Wang L, Wee Keong Ng, Bin Song (2011) Extended DCSP approach on product configuration with cost estimation. *Concurrent Engineering Research and Applications* 19(2):123–138
- Wang L, Zhong SS, Zhang YJ (2017a) Process configuration based on generative constraint satisfaction

- problem. *Journal of Intelligent Manufacturing* 28(4):945–957
- Wang P, Gong Y, Xie H, Liu Y, Nee AY (2017b) Applying CBR to machine tool product configuration design oriented to customer requirements. *Chinese Journal of Mechanical Engineering (English Edition)* 30(1):60–76
- Wang W, Koren Y (2012) Scalability planning for reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 31(2):83–91
- Wang Y, Ma HS, Yang JH, Wang KS (2017c) Industry 4.0: a way from mass customization to mass personalization production. *Advances in Manufacturing* 5(4):311–320
- Wei W, Fan W, Li Z (2014) Multi-objective optimization and evaluation method of modular product configuration design scheme. *International Journal of Advanced Manufacturing Technology* 75(9–12):1527–1536
- Whitfield RI, Smith JS, Duffy AB (2002) Identifying Component Modules. In: *Artificial Intelligence in Design '02*. Kluwer Academic Publishers, Dordrecht, pp 571–592
- Xie H, Henderson P, Kernahan M (2005) Modelling and solving engineering product configuration problems by constraint satisfaction. *International Journal of Production Research* 43(20):4455–4469
- Xu Z, Liang M (2005) Concurrent Optimization of Product Module Selection and Assembly Line Configuration: A Multi-Objective Approach. *Journal of Manufacturing Science and Engineering* 127(4):875–884
- Xu Z, Liang M (2006) Integrated planning for product module selection and assembly line design/reconfiguration. *International Journal of Production Research* 44(11):39–65
- Yang D, Dong M (2013) Applying constraint satisfaction approach to solve product configuration problems with cardinality-based configuration rules. *Journal of Intelligent Manufacturing* 24(1):99–111
- Yang D, Dong M (2012) A hybrid approach for modeling and solving product configuration problems. *Concurrent Engineering Research and Applications* 20(1):31–42
- Yang D, Dong M, Chang XK (2012) A dynamic constraint satisfaction approach for configuring structural products under mass customization. *Engineering Applications of Artificial Intelligence* 25(8):1723–1737
- Yang D, Li J, Wang B, Jia YJ (2020) Module-Based product configuration decisions considering both economical and carbon emission-related environmental factors. *Sustainability (Switzerland)* 12(3):1–13
- Yang D, Li X, Jiao RJ, Wang B (2018) Decision support to product configuration considering component replenishment uncertainty: A stochastic programming approach. *Decision Support Systems* 105:108–118
- Yang X-S (2010) *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons
- Ye H, Liang M (2006) *Simultaneous modular product scheduling and manufacturing cell*



- reconfiguration using a genetic algorithm. *Journal of Manufacturing Science and Engineering, Transactions of the ASME* 128(4):984–995
- Ye H, Liang M (2005) Solving the combined modular product scheduling and production cell reconfiguration problem: A GA approach with parallel chromosome coding. *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005 Proceedings* 2:1348–1355
- Yigit AS, Allahverdi A (2003) Optimal selection of module instances for modular products in reconfigurable manufacturing systems. *International Journal of Production Research* 41(17):4063–4074
- Yigit AS, Ulsoy AG, Allahverdi A (2002) Optimizing modular product design for reconfigurable manufacturing. *Journal of Intelligent Manufacturing* 13(4):309–316
- Youssef AMA, ElMaraghy HA (2008) Availability consideration in the optimal selection of multiple-aspect RMS configurations. *International Journal of Production Research* 46(21):5849–5882
- Youssef AMA, ElMaraghy HA (2007) Optimal configuration selection for Reconfigurable Manufacturing Systems. *International Journal of Flexible Manufacturing Systems* 19(2):67–106
- Youssef AMA, ElMaraghy HA (2006) Modelling and optimization of multiple-aspect RMS configurations. *International Journal of Production Research* 44(22):4929–4958
- Zhang LL (2014) Product configuration: A review of the state-of-the-art and future research. *International Journal of Production Research* 52(21):6381–6398
- Zheng C, Le Duigou J, Bricogne M, Eynard B (2016) Multidisciplinary interface model for design of mechatronic systems. *Computers in Industry* 76:24–37
- Zheng P, Xu X, Yu S, Liu C (2017) Personalized product configuration framework in an adaptable open architecture product platform. *Journal of Manufacturing Systems* 43:422–435
- Zhu B, Wang Z, Yang H, Mo R, Zhao Y (2008) Applying fuzzy multiple attributes decision making for product configuration. *Journal of Intelligent Manufacturing* 19(5):591–598

## PUBLICATIONS

The publications resulting from this work are listed as following:

### Peer reviewed journal:

RC Sabioni, J Daaboul, J Le Duigou (2021) Concurrent optimization of modular product and Reconfigurable Manufacturing System configuration: a customer-oriented offer for mass customization. *International Journal of Production Research*. Doi: 10.1080/00207543.2021.1886369

RC Sabioni, J Daaboul, J Le Duigou (2021) An integrated approach to optimize the configuration of mass-customized products and reconfigurable manufacturing systems. *International Journal of Advanced Manufacturing Technology*. In press.

### Conferences:

#### I- International:

IF Paes, RC Sabioni, J Daaboul, J Le Duigou (2018) Product and process modular design : a review. In: 8<sup>th</sup> *International Conference on Mass Customization and Personalization*, Community of Europe, Novi Sad.

RC Sabioni, J Daaboul, J Le Duigou (2020) Optimization of Reconfigurable Manufacturing Systems configuration: a literature review. In: *International Joint Conference on Mechanics, Design Engineering and Advanced Manufacturing (JCM)*, Aix-en-Provence.

RC Sabioni, J Daaboul, J Le Duigou (2020) Optimal integration of product configuration and process planning to meet individual customer requirements in mass customization. In: *13th International Conference on Modeling, Optimization and Simulation (MOSIM)*, Agadir.

RC Sabioni, L Wartelle, J Daaboul, J Le Duigou (2021) Attribute-based integrated product process configurator for mass customization. In: 9th CIRP Global Web Conference – Sustainable, resilient, and agile manufacturing and service operations: Lessons from COVID-19.

#### II- National :

RC Sabioni, J Daaboul, J Le Duigou (2019) Ontology development to support integrated modular design for Mass Customization. In: *16ème Colloque National S.mart*, Karellis

## APPENDICES

### APPENDICE A – Parameters used for the small example.

A - Table 1 - Small example - Function compatibility

Instances	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
<b>M11</b>	1	0	0	1	0	0	0	0	0	0	0
<b>M12</b>	0	1	0	1	0	0	0	0	0	0	0
<b>M13</b>	0	0	1	1	0	0	0	0	0	0	0
<b>M21</b>	0	0	0	0	1	1	0	0	0	0	0
<b>M22</b>	0	0	0	0	0	1	0	0	0	0	0
<b>M31</b>	0	0	0	0	0	0	0	1	0	0	0
<b>M32</b>	0	0	0	0	0	0	1	0	0	0	0
<b>M41</b>	0	0	0	0	0	0	0	0	1	0	0
<b>M42</b>	0	0	0	0	0	0	0	0	1	1	0
<b>M43</b>	0	0	0	0	0	0	0	0	0	1	1

A - Table 2 - Small example - Module instances compatibility

Instances	M11	M12	M13	M21	M22	M31	M32	M41	M42	M43
<b>M11</b>	1	0	0	1	1	1	1	1	1	1
<b>M12</b>	0	1	0	1	1	1	1	1	1	1
<b>M13</b>	0	0	1	1	1	1	1	1	1	1
<b>M21</b>	1	1	1	1	0	1	1	0	1	1
<b>M22</b>	1	1	1	0	1	0	1	1	0	1
<b>M31</b>	1	1	1	1	0	1	0	1	1	1
<b>M32</b>	1	1	1	1	1	0	1	1	1	1
<b>M41</b>	1	1	1	0	1	1	1	1	0	0
<b>M42</b>	1	1	1	1	0	1	1	0	1	0
<b>M43</b>	1	1	1	1	1	1	1	0	0	1

A - Table 3 - Small example - Required operations

Module	M11	M12	M13	M21	M22	M31	M32	M41	M42	M43
1	1	1	1							
2	1		1							
3		1								
4	1		1							
5					1					
6				1						
7				1	1					
8				1						
9						1				
10						1	1			
11							1			
12						1	1			
13										1
14								1	1	
15										1
16								1	1	1

A - Table 4 - Small example - Material cost

Instances	Cost
M11	2
M12	2
M13	3,5
M21	8,76
M22	6,1
M31	15,9
M32	26,7
M41	16
M42	18,3
M43	17,8

A - Table 5 - Small example - Operation cost

Operations	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W4C5
1	30	26,5				33								
2			10		15				12,6					
3						12			14	16				
4			17,2				19					16		
5	19							17		23				
6				17,5		15		12			14			
7		15			12									
8				13						19	17,3			
9									22,8			24,9		34,6
10	7			40	30					60	54,8			
11								7						
12							6	9,7						
13			23,4				28,7		31					
14									7,64				9,5	
15									17,4				18,2	
16								13						18,9

A - Table 6 - Small example - Operation time

Operations	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W4C5
1	0,05	0,04				0,07								
2			0,02		0,09				0,14					
3						0,04			0,07	0,11				
4			0,04				0,16					0,11		
5	0,05							0,12		0,05				
6				0,07		0,09		0,04			0,07			
7		0,09			0,04									
8				0,05						0,16	0,12			
9									0,10			0,07		0,05
10	0,12			0,07	0,05					0,11	0,16			
11								1,10						
12							0,98	0,89						
13			0,05				0,02		0,08					
14									0,71				0,35	
15									0,35				0,18	
16								0,50						0,20

A - Table 7 - Small example - Configuration change cost

Machines	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W4C5
W1C1	0	0,7	0,8	1,3										
W1C2	0,8	0	1,2	0,9										
W1C3	0,6	1,3	0	0,7										
W1C4	1,1	0,8	1	0										
W2C1					0	0,9	1,1							
W2C2					1	0	0,6							
W2C3					0,7	0,4	0							
W3C1								0	1,4					
W3C2								1,1	0					
W4C1										0	0,5	0,7	1,1	0,38
W4C2										1,3	0	0,3	1,4	0,486
W4C3										0,8	1,2	0	0,85	0,71
W4C4										0,9	1	0,92	0	0,923
W4C5										0,378	0,7	0,6	1,36	0

A - Table 8 - Small example - Configuration change time

Machines	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W4C5
W1C1	0	29	21	3										
W1C2	7	0	10	28										
W1C3	20	29	0	15										
W1C4	18	15	18	0										
W2C1					0	23	23							
W2C2					11	0	9							
W2C3					22	11	0							
W3C1								0	13					
W3C2								13	0					
W4C1										0	24	6	22	29
W4C2										19	0	20	30	17
W4C3										4	10	0	5	31
W4C4										14	10	25	0	18
W4C5										28	14	27	13	0

A - Table 9 - Small example - Machine displacement cost

Machine	Cost
W1	0,64
W2	0,72
W3	0,9
W4	0,53

A - Table 10 - Small example - Machine displacement time

Machine	Time
W1	45
W2	37
W3	26
W4	32

A - Table 11 - Small example - Operation precedence matrix

Operation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0
8	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
9	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0
10	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
14	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
15	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

## APPENDICE B – Parameters used for the smartphone case study.

B - Table 1 - Functions and module instances compatibility.

Instances	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26
M11	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M12	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M13	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M21	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M22	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M23	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M24	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M31	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M32	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M33	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M34	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
M42	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
M51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
M52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
M61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
M62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
M71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
M72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
M73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

B - Table 2 - Module instances compatibility.

Instances	M11	M12	M13	M21	M22	M23	M24	M31	M32	M33	M34	M41	M42	M51	M52	M61	M62	M71	M72	M73
M11	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M12	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M13	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M21	0	1	0	1	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1
M22	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1
M23	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
M24	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M31	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1
M32	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1
M33	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1
M34	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
M41	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
M42	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
M51	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
M52	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
M61	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
M62	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
M71	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
M72	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0
M73	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1

B - Table 3 - Raw material cost of module instances.

Instances	Cost
M11	2
M12	2
M13	5
M21	52
M22	37
M23	16
M24	0
M31	5
M32	3
M33	8
M34	0
M41	15
M42	30
M51	24
M52	11
M61	68
M62	37

B - Table 4 - Distances between each pair of layout locations.

Machines	L1	L2	L3	L4	L5	L6	L7
L1	0	1	2	3	4	5	6
L2	1	0	1	2	3	4	5
L3	2	1	0	1	2	3	4
L4	3	2	1	0	1	2	3
L5	4	3	2	1	0	1	2
L6	5	4	3	2	1	0	1
L7	6	5	4	3	2	1	0



B - Table 5 - Required operations per module instance.

MODULE	M11	M12	M13	M21	M22	M23	M24	M31	M32	M33	M34	M41	M42	M51	M52	M61	M62
1	1	1	1														
2		1	1														
3	1		1														
4				1	1	1	1										
5				1	1	1											
6				1													
7					1												
8						1											
9								1	1	1	1						
10								1		1							
11								1		1							
12									1	1							
13									1	1							
14										1							
15												1	1				
16												1					
17													1				
18														1	1		
19														1			
20														1	1		
21															1		
22																1	1
23																1	
24																1	1
25																	1

**B - Table 6. Time t of operation p being processed by machine w with configuration c.**

Operations	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4	W5C5	W6C1	W6C2	W6C3	W7C1	W7C2
1	0,6	1,2																					
2	0,8																						
3			0,3																				
4				0,9			0,5																
5					0,8	1,1																	
6				0,6	0,2																		
7							0,4	0,6															
8					0,4			0,2															
9							1,6			1,2											0,7		
10								1,2	0,9														
11										0,5	0,7									0,15			
12									1,2		1,1												
13									0,7		1,6												
14												2,30								0,3			
15								1,6	0,9														
16											0,3												
17									0,1			0,20											
18													2,40		2,55								
19														1,80		1,70							
20																		0,20					
21															2,40		0,34		1,1				
22						0,4			0,6														
23								1,1			0,7												0,12
24										0,9		0,16											
25								0,5		0,45													
26									5,1		0,56	0,72											
27								0,6		0,41	0,22												0,92
28							1		0,8		0,7												
29										0,3		0,20										0,3	
30									0,8		0,1												
31															0,98			0,6			0,8		
32																				1,3		0,15	

**B - Table 7. Cost c of operation p being processed by machine w with configuration c.**

Operations	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4	W5C5	W6C1	W6C2	W6C3	W7C1	W7C2	
1	19,00	31,00																						
2	31,00																							
3			25,00																					
4				28,00					33,00															
5					22,00	34,00																		
6				22,00	3,00																			
7							26,00	29,00																
8					23,00				25,00															
9							2,00			24,00											3,00			
10								23,00	32,00															
11										18,00	28,00									33,00				
12										25,00	22,00													
13										34,00	19,00													
14												25,00									27,00			
15										16,00	22,00													
16											2,00													
17									21,00			32,00												
18													19,00		18,00									
19														27,00		33,00								
20																	19,00							
21															3,00		21,00		28,00					
22						31,00				28,00														
23								27,00			33,00													17
24										25,00		19,00												
25									32,00	26,00														
26										3,00	29,00	26,00												
27								2,00		32,00	18,00													29
28							26,00		29,00	33,00														
29										33,00		18,00										31,00		
30									18,00		24,00													
31															13,00			33,00			23,00			
32																					28,00		31,00	

**B - Table 8 - Cost (cu) per time unit (tu) of displacing machines and time per distance unit of displacing machine.**

Machines	Cost	Time
W1	0,34	20
W2	0,72	37
W3	0,49	26
W4	0,53	32
W5	0,42	28
W6	0,89	21
W7	0,76	19

**B - Table 9 - Cost of changing machine's configuration.**

	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4	W5C5	W6C1	W6C2	W6C3	W7C1	W7C2	
W1C1	0	0,7	0,8	1,3																				
W1C2	0,8	0	1,2	0,9																				
W1C3	0,6	1,3	0	0,7																				
W1C4	1,1	0,8	1	0																				
W2C1					0	0,9	1,1																	
W2C2					1	0	0,6																	
W2C3					0,7	0,4	0																	
W3C1								0	1,4															
W3C2								1,1	0															
W4C1										0	0,5	0,7	1,1											
W4C2										1,3	0	0,3	1,4											
W4C3										0,8	1,2	0	0,85											
W4C4										0,9	1	0,92	0											
W5C1														0	0,6	0,37	0,55	0,96						
W5C2														0,8	0	0,43	0,11	1						
W5C3														0,25	9	0	0,13	1,11						
W5C4														0,36	0,94	0,3	0	1,3						
W5C5														0,9	1,1	1,2	1,3	0						
W6C1																			0	0,35	0,52			
W6C2																			0,15	0	0,38			
W6C3																			0,28	0,47	0			
W7C1																						0	0,38	
W7C2																						0,61	0	

**B - Table 10 - Time of changing machine's configuration.**

	W1C1	W1C2	W1C3	W1C4	W2C1	W2C2	W2C3	W3C1	W3C2	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4	W5C5	W6C1	W6C2	W6C3	W7C1	W7C2	
W1C1	0	29	21	3																				
W1C2	7	0	10	28																				
W1C3	20	29	0	15																				
W1C4	18	15	18	0																				
W2C1					0	23	23																	
W2C2					11	0	9																	
W2C3					22	11	0																	
W3C1								0	13															
W3C2								13	0															
W4C1										0	24	6	22											
W4C2										19	0	20	30											
W4C3										4	10	0	5											
W4C4										14	10	25	0											
W5C1														0	6	7	5,5	9,6						
W5C2														8	0	11	11,5	10,7						
W5C3														5	9	0	13	11,1						
W5C4														6,3	9,4	10	0	12,3						
W5C5														8,2	11	12	13	0						
W6C1																			0	7,2	8,5			
W6C2																			8,15	0	9,8			
W6C3																			6,8	7,4	0			
W7C1																						0	8,3	
W7C2																						9	0	

## APPENDICE C – Parameters used for the sneaker case study.

C - Table 1 - Shoe example - Function compatibility

Instances	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28
M11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
M12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
M13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
M21	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M22	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M31	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
M32	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
M33	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
M34	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
M41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
M42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
M43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
M51	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
M52	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
M53	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
M54	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
M55	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
M61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
M62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
M63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

C - Table 2 - Shoe example - Module instance compatibility

Instances	M11	M12	M13	M21	M22	M31	M32	M33	M34	M41	M42	M43	M51	M52	M53	M54	M55	M61	M62	M63
M11	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M12	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M13	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1
M21	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M22	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
M31	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1
M32	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	0	1	1	1	1
M33	1	1	0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1
M34	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
M41	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1
M42	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
M43	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
M51	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0
M52	1	1	1	1	1	1	1	0	0	1	1	1	0	1	0	0	0	1	1	1
M53	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	0	0	1	1	0
M54	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	0
M55	1	1	1	1	1	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1
M61	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
M62	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
M63	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1

C - Table 3 - Shoe example - Required operations

Module	M11	M12	M13	M21	M22	M31	M32	M33	M34	M41	M42	M43	M51	M52	M53	M54	M55	M61	M62	M63
1		1	1																	
2	1		1																	
3	1	1	1																	
4				1	1															
5				1																
6				1																
7					1															
8				1																
9					1															
10						1	1		1											
11						1														
12						1		1												
13							1	1	1											
14							1	1	1											
15							1													
16							1		1											
17										1	1									
18										1	1	1								
19										1		1								
20										1	1									
21													1			1	1			
22														1	1	1				
23													1	1		1				
24														1	1		1			
25													1		1	1	1			
26													1			1				
27															1		1			
28																		1	1	1
29																				1
30																		1	1	
31																		1	1	1
32																				1
33																			1	1

C - Table 4 - Shoe example - Material cost

Instances	M11	M12	M13	M21	M22	M31	M32	M33	M34	M41	M42	M43	M51	M52	M53	M54	M55	M61	M62	M63
Cost	0,043	1	2	6	3	3,5	3,5	2,5	2,5	0,5	0,1	0,3	20,65	10,75	5,34	1,43	0,64	0,8	1,75	1

C - Table 5 - Shoe example - Operation cost

Operations	W1C1	W1C2	W1C3	W1C4	W1C5	W2C1	W2C2	W2C3	W3C1	W3C2	W3C3	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4
1	16			18				19											
2			25			15		26					23						
3		21				21				15				17				15	
4				25			22		23			19					24		
5		19						15			26				16		17		
6	20									21								14	
7			24	25												26			
8					27														16
9							27				19					19			
10		19				16			20	16									
11													23		19				
12														21			19		
13																15	27		
14			20				27												
15											17			16					
16		15	17						22										
17						27				24		15							18
18		16												24					
19	25			19								27							
20							16		20							27		18	
21													26		24				
22	27				14			23											
23				18											16		19		
24											25					15			16
25	14							24											
26											20								21
27							17						26				18		
28					16					26									
29															25				16
30									23		27								
31															15			18	
32			16			25													
33													23		14				

C - Table 6 - Shoe example - Operation time

Ops.	W1C1	W1C2	W1C3	W1C4	W1C5	W2C1	W2C2	W2C3	W3C1	W3C2	W3C3	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4
1	0,02			0,03				0,03											
2			0,04			0,05		0,07					0,03						
3		0,04				0,04				0,06				0,05				0,05	
4				0,01			0,04		0,05			0,04					0,06		
5		0,08						0,08			0,07				0,07		0,07		
6	0,10									0,11								0,11	
7			0,12	0,02												0,09			
8					0,02														0,04
9							0,03				0,03					0,03			
10		0,03				0,06			0,06	0,03									
11													0,08		0,07				
12														0,07			0,08		
13																	0,07	0,07	
14			0,06				0,05												
15											0,06			0,07					
16		0,02	0,04						0,04										
17						0,05				0,02		0,05							0,03
18		0,01												0,04	0,03				
19	0,14			0,14								0,13							
20							0,10		0,08							0,09		0,07	
21													0,03		0,07				

C - Table 7 - Shoe example - Configuration change cost

Machines	W1C1	W1C2	W1C3	W1C4	W1C5	W2C1	W2C2	W2C3	W3C1	W3C2	W3C3	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4
W1C1	0,000	0,068	0,454	0,024	0,341														
W1C2	0,518	0,000	0,225	0,834	0,021														
W1C3	0,540	0,546	0,000	0,447	0,109														
W1C4	0,691	0,693	0,885	0,000	0,464														
W1C5	0,582	0,578	0,163	0,309	0,000														
W2C1						0,000	0,866	0,765											
W2C2						0,333	0,000	0,988											
W2C3						0,641	0,435	0,000											
W3C1									0,000	0,876	0,595								
W3C2									0,266	0,000	0,408								
W3C3									0,645	0,336	0,000								
W4C1												0,000	0,268	0,365	0,549				
W4C2												0,209	0,000	0,030	0,410				
W4C3												0,335	0,195	0,000	0,189				
W4C4												0,260	0,264	0,918	0,000				
W5C1																0,000	0,253	0,966	0,389
W5C2																0,979	0,000	0,546	0,731
W5C3																0,308	0,442	0,000	0,331
W5C4																0,071	0,482	0,626	0,000

C - Table 8 - Shoe example - Configuration change time

Machines	W1C1	W1C2	W1C3	W1C4	W1C5	W2C1	W2C2	W2C3	W3C1	W3C2	W3C3	W4C1	W4C2	W4C3	W4C4	W5C1	W5C2	W5C3	W5C4
W1C1	0	18	10	18	8														
W1C2	13	0	8	12	14														
W1C3	12	7	0	18	19														
W1C4	16	10	17	0	14														
W1C5	9	19	5	11	0														
W2C1						0	18	11											
W2C2						6	0	13											
W2C3						11	13	0											
W3C1									0	12	10								
W3C2									5	0	15								
W3C3									12	10	0								
W4C1												0	11	20	17				
W4C2												11	0	13	12				
W4C3												7	20	0	18				
W4C4												15	5	9	0				
W5C1																0	11	18	9
W5C2																19	0	13	9
W5C3																16	6	0	10
W5C4																8	16	13	0

C - Table 9 - Shoe example - Machine displacement cost

Machine	Cost
W1	0,14
W2	0,23
W3	0,41
W4	0,35
W5	0,24

C - Table 10 - Shoe example - Machine displacement time

Machine	Time
W1	10
W2	23
W3	16
W4	22
W5	18



C - Table 11 - Shoe example - Operation precedence matrix

Operations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0